10 of 10

Signetics

Microprocessor Data Manual

Signetics

Microprocessor Data Manual

Signetics

Microprocessor Products

Data
Manual
1987

Microprocessor Products

Signetics reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Signetics assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustrative purposes only. Signetics makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

LIFE SUPPORT APPLICATIONS

Signetics' Products are not designed for use in life support appliances, devices, or systems where malfunction of a Signetics Product can reasonably be expected to result in a personal injury. Signetics' customers using or selling Signetics' Products for use in such applications do so at their own risk and agree to fully indemnify Signetics for any damages resulting in such improper use or sale.

Signetics registers eligible circuits under
the Semiconductor Chip Protection Act.

© Copyright 1987 Signetics Corporation

All rights reserved.

Signetics

Contents

Microprocessor Products

Ordering Information v

Product Status vi

Section 1 – QUALITY AND RELIABILITY

Quality and Reliability 1-3

Section 2 – MICROPROCESSOR PRODUCTS

SCN2641 Asynchronous Commmunications Interface 2-3
SCN2651 Programmable Communications Interface (PCI) 2-16
SCN2652/68652 Multi-Protocol Communications Controller (MPCC) 2-32
SCN2653/68653 Polynomial Generator Checker (PGC) 2-52
SCN2661/68661 Enhanced Programmable Communications Interface (EPCI) 2-70
SCN2671 Programmable Keyboard and Communication Controller (PKCC) 2-87
SCN2672 Programmable Video Timing Controller (PVTC) 2-109
SCB2673 Video Attributes Controller (VAC) 2-132
SCN2674 Advanced Video Display Controller (AVDC) 2-145
SCB2675 Color/Monochrome Attributes Controller (CMAC) 2-177
SCB2675T Turbo Color/Monochrome Attributes Controller (Turbo-CMAC) 2-188
SCB2677 Video Attributes Controller (VAC) 2-199
SCN2681 Dual Asynchronous Receiver/Transmitter (DUART) 2-211
SCC2691 Universal Asynchronous Receiver/Transmitter (UART) 2-230
SCC2692 Dual Asynchronous Receiver/Transmitter (DUART) 2-247
SCC2698 Octal Universal Asynchronous Receiver/Transmitter (Octal UART) 2-266
SCC63484 Advanced CRT Controller (ACRTC) 2-285
SCN68000 16-/32-Bit Microprocessor 2-340
SCN68010 16-Bit Virtual Memory Microprocessor 2-402
SCB68154 Interrupt Generator 2-471
SCB68155 Interrupt Handler 2-482
SCB68171 Very Little Serial Interface Chip (VLSIC) 2-499
SCB68172 VMEbus Controller (BUSCON) 2-504
SCC68173 VMSbus Controller (VMSCON) 2-530
SCB68430 Direct Memory Access Interface (DMAI) 2-544
SCN68562 Dual Universal Serial Communications Controller (DUSCC) 2-565
SCN68681 Dual Asynchronous Receiver/Transmitter (DUART) 2-612
SCC68692 Dual Asynchronous Receiver/Transmitter (DUART) 2-632

Section 3 – MICROCONTROLLER PRODUCTS

SCN8049 Series Single-Chip 8-Bit Microcontroller 3-3
SCN8031AH/SCN8051AH Single-Chip 8-Bit Microcontroller 3-18
SCN8032AH/SCN8052AH Single-Chip 8-Bit Microcontroller 3-29
SC80C31B/SC80C51B CMOS Single-Chip 8-Bit Microcontroller 3-40
SC80C451/SC83C451 I/O Expanded CMOS Microcontroller 3-54
SC87C451 CMOS Single-Chip 8-Bit EPROM Microcontroller 3-70
SC83C751 CMOS Single-Chip 8-Bit Microcontroller 3-71
SC80C552/SC83C552 CMOS Automotive and Industrial Microcontroller 3-73
SC80C652/SC83C652 CMOS Single-Chip 8-Bit Microcontroller 3-76
SC96AH Series Single-Chip 16-Bit Microcontrollers 3-79
SC96BH Series Single-Chip 16-Bit Microcontrollers 3-103
PCB5010/11 Single-Chip Digital Signal Processors 3-106
I^2C Bus Specification 3-111
AN408 SC83C451 Operation of Port 6 3-123
8X305 Microcontroller 3-134
8X310 Interrupt Control Coprocessor 3-159
8X320 Bus Interface Register Array 3-172
8X330 Floppy Disk Formatter/Controller 3-180
8X353 Bipolar RAM (32 × 8) 3-197

Contents

8X355 LIFO Stack Memory (32 × 8) 3-205
8X360 Memory Address Director (MAD) 3-214
8X371 8-Bit Latched Bidirectional I/O Port 3-223
8X372/8X376 Addressable/Bidirectional I/O Port 3-230
8X374 Addressable/Bidirectional I/O Port with Parity 3-239
8X382 4-Input/4-Output Addressable I/O Port 3-249
8X401 Microcontroller 3-259
AN4011 An 8X401-Based Raster Graphics System 3-279
8X411 Priority Interrupt Encoder 3-304
8X450 RAM 3-306
8X470 I/O Port 3-313
MCCAP 8X300/8X305 Cross Assembler Program 3-327
FORTRESS 8X401 Cross Assembler Program 3-328
8X300 KT1SK 8X305 Prototyping System 3-329
8X305 Development System American Automation EZ-PRO 3-331
I/O Port Programmer 3-332

Section 4 - PACKAGE OUTLINES
A Plastic Leaded Chip Carrier 4-3
D Plastic Small Outline 4-6
F Hermetic CERDIP 4-8
I Hermetic SDIP 4-9
N Plastic DIP 4-13
P Grid Array 4-18

Section 5 - SALES OFFICES
Sales Offices 5-3

Section 6 - ALPHANUMERIC INDEX
Alphanumeric Index 6-3

Signetics

Ordering Information

Microprocessor Products

PART NUMBERING SYSTEM

Example: **S C N 6 8 4 5 4 C 6 N 4 8**

Microprocessor Division
Identifier – Always SC

Process/Power Variation
N = N – Channel
C = C – MOS*
B = Bipolar

Basic Part Number
See individual data sheets.

Pin Count
14, 16, 20, 24, 28, 40, 48, etc.

Package
I = Ceramic
N = Plastic
F = Cerdip
A = PLCC
P = PGA

Temperature
C = 0 C to 70 C
(Commercial)
A = −40 C to +85 C
(Automotive)
M = −55 C to +125 C
(Military)
P = −20 C to +70 C
(Philips)

Timing Variation

Spd	Sym	Spd	Sym	Spd	Sym
01	1	21	1	41	1
02	2	22	2	42	2
03	3	23	3	43	3
04	4	24	4	44	4
05	5	25	5	45	5
06	6	26	6	46	6
07	7	27	7	47	7
08	8	28	8	48	8
09	9	29	9	49	9
10	A	30	0	50	0
11	B	31	1	51	1
12	C	32	2	52	2
13	D	33	3	53	3
14	E	34	4	54	4
15	F	35	5	55	5
16	6	36	6	56	6
17	7	37	7	57	7
18	8	38	8	58	8
19	9	39	9		
20	0	40	0		

Unless otherwise noted.

---

*CMOS designed may be part of basic part type.

Product Status

DEFINITIONS		
Data Sheet Identification	**Product Status**	**Definition**
Objective Specification	**Formative or In Design**	This data sheet contains the design target or goal specifications for product development. Specifications may change in any manner without notice.
Preliminary Specification	**Preproduction Product**	This data sheet contains preliminary data and supplementary data will be published at a later date. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product Specification	**Full Production**	This data sheet contains Final Specifications. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

Signetics

Microprocessor Products

Section 1
Quality and Reliability

Quality and Reliability

Microprocessor Products

INTRODUCTION

The Microprocessor Division was formed in 1979 when it became apparent that microprocessor circuits and their peripheral devices would become increasingly important system components to the OEM customer base. Because of performance to cost ratios, reduced power requirements and inherent reliability, these components provide the solution to designers' complex system requirements.

NMOS became the dominant technology because of the simplicity of its design and processing and its large historical reliability data base. To date, NMOS has serviced 95% of the division's device requirements. Bipolar devices are used to a limited extent as interface drivers between the NMOS system components and CRT displays.

The Microprocessor Division serves a broad base of customers with its product lines of eight and sixteen bit microprocessors, microcontrollers, and data communication and CRT peripheral circuits. The division's customers range from large mainframe computer manufacturers to small systems users, and the division can generally satisfy all the microprocessor requirements of any customer.

Our goal is to establish ourselves as the preferred alternate source in both the 68000 microprocessor and the 80XX microcontroller product areas and at the same time become the prime supplier of the interface and peripheral circuits required to integrate the parts within a system. Our design philosophy is to give superior performance while minimizing the overall number of parts required for the overall system.

The terms quality and reliability are often misinterpreted. In general, quality refers to the condition of a device when received; reliability covers extent of useful life. Quality is readily measurable; reliability is predictable and verifiable based on historical evidence.

MICROPROCESSOR DIVISION RELIABILITY

No amount of stress testing can improve a product's reliability. Stress tests are used to measure and define an end of life which can be expected from a family of products. Reliability as well as quality must be built in through proper design, processing, assembly, testing and handling. For Signetics' Microprocessor Division parts, accelerated life test data show an extremely reliable product line and serve as one input for continuous product reliability improvement.

DESIGN

Product quality and reliability begin in design. Strategic questions directly affecting reliability must be answered. How much static protection is required on input leads? Do formal design rules exist? Can they ever be violated? What method is used to anticipate future processes or "shrinks"?

Within Signetics' Microprocessor Division, rigid guidelines are in effect to ensure compliance with our design rules. Design rules, once established, are inviolate.

QUALIFICATION TESTING

Signetics' Microprocessor Division verifies device reliability through a series of qualification tests and a continuous reliability monitor program, Sure III (Systematic Uniform Reliability Evaluation).

All new fab processes at Signetics are qualified by stress testing parts from a variety of production lots. This accelerated stress testing is shown in the table.

SURE III

Continuous reliability monitoring is performed via our SURE III program. Devices are randomly selected from production lots and subjected to the same environmental stresses noted in the table. The program is administered by the Corporate Reliability Engineering Group, which publishes a summary of results on a quarterly basis.

The SURE III program covers two functions: Monitoring short term and long term reliability performance.

LONG-TERM AUDIT

One hundred devices from each generic family are subjected to each of the following stresses every other four weeks:

- High Temperature Operating Life — $T_J = 150°C$, 1000 hours — (Static Biased or Dynamic Operation, as appropriate);
- Temperature-Humidity Biased Life — 85°C, 85% RH, 1000 hours, static biased;
- Temperature Cycling (Air-Air) — −65°C to +150°C, 1000 cycles.

SHORT-TERM MONITOR

Every week 20-piece samples from each generic family are run to 96 hours of pressure pot (15 psig, 121°C, 100% saturated steam), 300 cycles of thermal shock (−65°C to +150°C) and 168 hours of high temperature operating life ($T_J = 150°C$, static or dynamic operation).

In addition, each Signetics assembly plant performs SURE product monitor stresses weekly on each generic family and molded package, by pin count and frame type. Fifty pieces are subjected to 300 cycles of thermal shock (Cond. C) and 100 devices are subjected to pressure pot stress at 20 psig for 72 hours (168 hour equivalent at 15 psig).

Quality and Reliability

Accelerated Life Stress Tests

TEST		TEST CONDITION	NUMBER OF DEVICES	DURATION
DHTL	Dynamic high temperature life	T_A@125°C (operating)	52	1000 hrs
SHTL	Static high temperature life	T_A@125°C (operating)	52	1000 hrs
THBS	Biased temperature humidity life	85°C/85% RH (operating)	52	1000 hrs
PPOT	Pressure pot (autoclave)	121°C/15 psig (storage)	52	96 hrs
TMSK	Thermal shock	−55°C/125°C liq. to liq. (storage)	77	300 cycles
TMCL	Temperature cycle (air-to-air)	−65°C/150°C (storage)	77	300 cycles

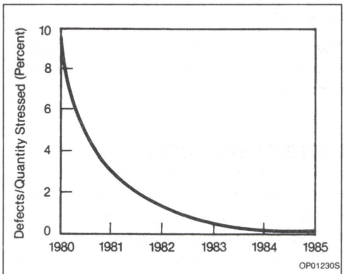


Figure 1. Microprocessor Reliability Index

Figure 1 shows the relative improvement in Sure III performance for the Microprocessor Division for the past five years. This curve combines all the stress tests and plots the overall percentage defective by year. Clearly, the reliability of our product line is improving, and more importantly, it is consistent and predictable.

SIGNETICS' QUALITY IMPROVEMENT PROGRAM

Signetics began its Quality Improvement Program in 1980, and developed it around the concepts outlined in the book, "Quality is Free," by Phil Crosby.

This program, which is actively supported by top management, defines Quality as "Conformance to the Specification." With this definition in mind, our performance standard is "Zero Defects." Tracking charts measuring quality improvement targets are used and displayed throughout the Division.

Microprocessor Division personnel are all actively involved in this program. Our people have taken formal training in Quality College and have pledged to "Do It Right the First Time, On Time." Administrative personnel promise to "Make Certain" of their own work.

The program improves quality through education, commitment and feedback.

Internally, there are many signs of the program's success. The SURE III Reliability Assurance Monitor shows improved results in each of the past three years. In-line quality improvements are impressive. All new products are placed on QRA Hold until completion of environmental stress testing; all significant process changes go through qualification prior to release to production; test programs are controlled and released only after extensive engineering correlation; wafers with less than the required minimum number of good die are scrapped to avoid jeopardizing product quality.

These are examples of how the Quality Improvement Program functions internally. However, the real measure of the program is customer belief in and acceptance of our improvement.

Over the past few years, customers have verified a continuous improvement in the quality and reliability of Microprocessor Division parts. Figure 2 shows the improvement in visual mechanical and electrical outgoing quality levels over the past several years.

In addition to the marked improvement in our AOQ (Average Outgoing Quality), we have become a qualified source for the 80XX and the 68000 families, as well as our own proprietary peripheral circuits, for more than 200 customers. We have developed Ship-to-Stock programs for several key accounts, eliminating the need for costly incoming inspection by our customers.

The improved reliability of our parts has allowed several customers to drop their requirements for burn-in for high reliability programs. This, of course, has lowered their costs significantly and has allowed them to improve their over-all system reliability.

Reliability testing is an important monitor of our manufacturing process. Signetics' microprocessor parts not only meet specifications when shipped but continue to operate satisfactorily throughout their lifetime.

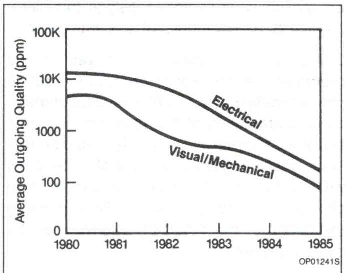


Figure 2. Outgoing Quality Level Microprocessor Division

Signetics

Section 2
Microprocessor Products

Microprocessor Products

INDEX

SCN2641 Asynchronous Commmunications Interface 2-3
SCN2651 Programmable Communications Interface (PCI) 2-16
SCN2652/68652 Multi-Protocol Communications Controller (MPCC) 2-32
SCN2653/68653 Polynomial Generator Checker (PGC) 2-52
SCN2661/68661 Enhanced Programmable Communications Interface (EPCI) 2-70
SCN2671 Programmable Keyboard and Communication Controller (PKCC) 2-87
SCN2672 Programmable Video Timing Controller (PVTC) 2-109
SCB2673 Video Attributes Controller (VAC) 2-132
SCN2674 Advanced Video Display Controller (AVDC) 2-145
SCB2675 Color/Monochrome Attributes Controller (CMAC) 2-177
SCB2675T Turbo Color/Monochrome Attributes Controller (Turbo-CMAC).... 2-188
SCB2677 Video Attributes Controller (VAC) 2-199
SCN2681 Dual Asynchronous Receiver/Transmitter (DUART).................. 2-211
SCC2691 Universal Asynchronous Receiver/Transmitter (UART).............. 2-230
SCC2692 Dual Asynchronous Receiver/Transmitter (DUART).................. 2-247
SCC2698 Octal Universal Asynchronous Receiver/Transmitter (Octal UART)... 2-266
SCC63484 Advanced CRT Controller (ACRTC) 2-285
SCN68000 16-/32-Bit Microprocessor... 2-340
SCN68010 16-Bit Virtual Memory Microprocessor 2-402
SCB68154 Interrupt Generator 2-471
SCB68155 Interrupt Handler 2-482
SCB68171 Very Little Serial Interface Chip (VLSIC) 2-499
SCB68172 VMEbus Controller (BUSCON) 2-504
SCC68173 VMSbus Controller (VMSCON) 2-530
SCB68430 Direct Memory Access Interface (DMAI) 2-544
SCN68562 Dual Universal Serial Communications Controller (DUSCC)........ 2-565
SCN68681 Dual Asynchronous Receiver/Transmitter (DUART).................. 2-612
SCC68692 Dual Asynchronous Receiver/Transmitter (DUART).................. 2-632

Signetics

Microprocessor Products

SCN2641
Asynchronous Communications Interface

Product Specification

DESCRIPTION

The Signetics SCN2641 is a universal asynchronous data communications controller chip that interfaces directly to most 8-bit microprocessors and may be used in a polled or interrupt-driven system environment. The SCN2641 accepts programmed instructions from the microprocessor while supporting asynchronous serial data communications in full- or half-duplex mode.

The SCN2641 serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The SCN2641 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode.

The SCN2641 is constructed using Signetics n-channel silicon gate depletion load technology and is packaged in a 24-pin DIP.

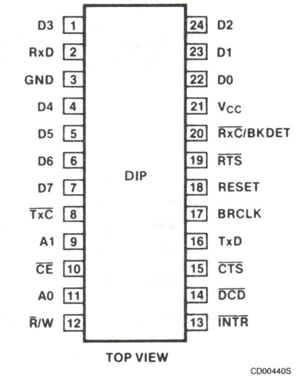
FEATURES

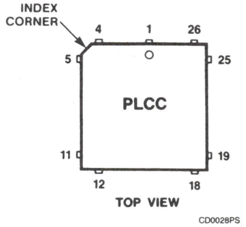
- **5- to 8-bit characters plus parity**
- **1, 1½ or 2 stop bits transmitted**
- **Odd, even or no parity**
- **Parity, overrun and framing error detection**
- **Line break detection and generation**
- **False start bit detection**
- **Automatic serial echo mode (echoplex)**
- **Local or remote maintenance loopback mode**
- **Baud rate:**
  - **DC to 1M bps (1X clock)**
  - **DC to 62.5K bps (16X clock)**
  - **DC to 15.625K bps (64X clock)**
- **Internal or external baud rate clock**
- **16 internal rates**
- **Double-buffered transmitter and receiver**
- **Single +5V power supply**
- **400 mil package width**

APPLICATIONS

- **Intelligent terminals**
- **Network processors**
- **Front-end processors**
- **Remote data concentrators**
- **Serial peripherals**

PIN CONFIGURATIONS


D3 1
RxD 2
GND 3
D4 4
D5 5
D6 6
D7 7
$\overline{\text{TxC}}$ 8
A1 9
$\overline{\text{CE}}$ 10
A0 11
$\overline{\text{R}}$/W 12
24 D2
23 D1
22 D0
21 V_{CC}
20 $\overline{\text{RxC}}$/BKDET
19 $\overline{\text{RTS}}$
18 RESET
17 BRCLK
16 TxD
15 $\overline{\text{CTS}}$
14 $\overline{\text{DCD}}$
13 $\overline{\text{INTR}}$
DIP
TOP VIEW
CD00440S

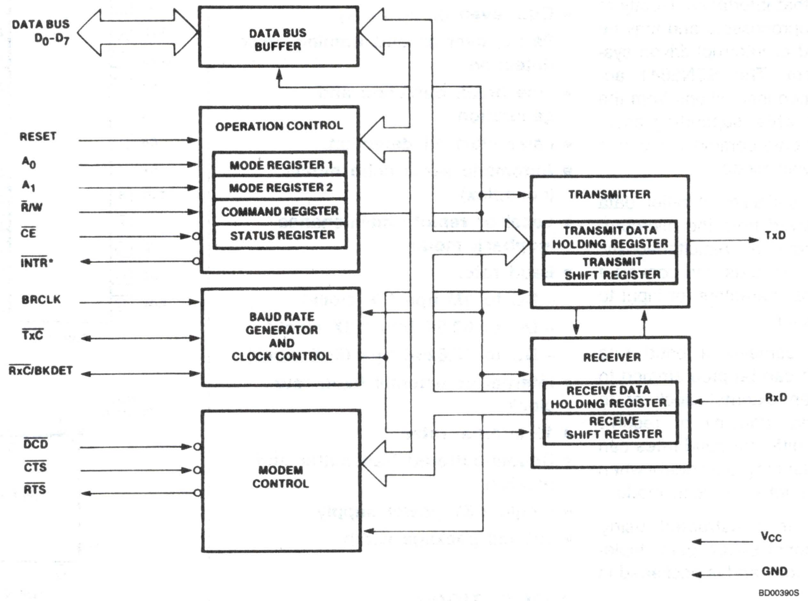

INDEX CORNER
4
1
26
5
25
PLCC
11
19
12
18
TOP VIEW
CD0028PS

Pin	Function	Pin	Function
1	D2	15	$\overline{\text{INTR}}$
2	D3	16	$\overline{\text{DCD}}$
3	RxD	17	$\overline{\text{CTS}}$
4	GND	18	NC
5	D4	19	TxD
6	D5	20	BRCLK
7	D6	21	RESET
8	NC	22	NC
9	D7	23	$\overline{\text{RTS}}$
10	$\overline{\text{TxC}}$	24	NC
11	A1	25	$\overline{\text{RxC}}$/BKDET
12	$\overline{\text{CE}}$	26	V_{CC}
13	A0	27	D0
14	$\overline{\text{R}}$/W	28	D1

ORDERING INFORMATION

PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to 70°C
Plastic DIP	SCN2641CC1N24
Plastic LCC	SCN2641CC1A28

BLOCK DIAGRAM


DATA BUS D_0-D_7
DATA BUS BUFFER
OPERATION CONTROL
MODE REGISTER 1
MODE REGISTER 2
COMMAND REGISTER
STATUS REGISTER
RESET
A_0
A_1
$\overline{R}$/W
$\overline{CE}$
$\overline{INTR}$*
BRCLK
$\overline{TxC}$
$\overline{RxC}$/BKDET
BAUD RATE GENERATOR AND CLOCK CONTROL
$\overline{DCD}$
$\overline{CTS}$
$\overline{RTS}$
MODEM CONTROL
TRANSMITTER
TRANSMIT DATA HOLDING REGISTER
TRANSMIT SHIFT REGISTER
TxD
RECEIVER
RECEIVE DATA HOLDING REGISTER
RECEIVE SHIFT REGISTER
RxD
V_{CC}
GND
BD00390S

NOTE:
*Open drain output pin

BLOCK DIAGRAM

The SCN2641 consists of five major sections: the transmitter, receiver, timing, operation control and modem control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers 1 and 2, the command register, and the status register. Details of register addressing are presented in the SCN2641 programming section of this data sheet.

Timing

The SCN2641 contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full-duplex operation. See Table 1.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for certain errors and sends an "assembled" character to the CPU.

Transmitter

The transmitter accepts parallel data from the CPU, appends start and stop bits, and, optionally, a parity bit, and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for two input signals and one output signal used for "handshaking" and status indication between the CPU and a modem.

INTERFACE SIGNALS

The SCN2641 interface signals can be grouped into two types: the CPU-related signals (shown in Table 2), which interface the SCN2641 to the microprocessor system and the device-related signals (shown in Table 3), which are used to interface to the communications device or system.

OPERATION

The functional operation of the SCN2641 is programmed by a set of control words supplied by the CPU. These control words speci-

Table 1. Baud Rate Generator Characteristics (BRCLK = 3.6864MHz)

MR23 – 20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	—	4608
0001	75	1.2	—	3072
0010	110	1.7596	-0.022	2095
0011	134.5	2.152	—	1713
0100	150	2.4	—	1536
0101	300	4.8	—	768
0110	600	9.6	—	384
0111	1200	19.2	—	192
1000	1800	28.8	—	128
1001	2000	32.055	0.174	115
1010	2400	38.4	—	96
1011	3600	57.6	—	64
1100	4800	76.8	—	48
1101	7200	115.2	—	32
1110	9600	153.6	—	24
1111	19200	307.2	—	12

fy items such as baud rate, number of bits per character, etc. The programming procedure is described in the SCN2641 programming section of this data sheet.

After programming, the SCN2641 is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The SCN2641 is conditioned to receive data when the $\overline{\text{DCD}}$ input is low and the RxEN bit in the command register is true. The receiver looks for a high-to-low transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and one stop bit have been assembled. The data is then transferred to the receive data holding register, the RxRDY bit in the status register is set, and the $\overline{\text{INTR}}$ output is asserted. If the character length is less than 8 bits, the high-order unused bits in the holding register are set to zero. The parity error, framing error, and overrun error status bits are strobed into the status register on the positive-going edge of $\overline{\text{RxC}}$ corresponding to the received character boundary. If the stop bit is present, the receiver will immediately begin its search for the next start bit. If the stop bit is absent (framing error), the receiver will interpret a space as a start bit if it persists into the next bit time interval. If a break condition is detected (RxD is low for the entire character as well as the stop bit), only one character consisting of all zeros (with the FE status bit set) will be transferred to the holding register. The RxD input must return to a high condition before a search for the next start bit begins.

Pin 20 can be programmed to be a break detect output by appropriate setting of MR27 – MR24. If so, a detected break will cause that pin to go high. When RxD returns to mark for one RxC time, pin 20 will go low. Refer to the break detection timing diagram.

Transmitter

The SCN2641 is conditioned to transmit data when the $\overline{\text{CTS}}$ input is low and the TxEN command register bit is set. The SCN2641 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{\text{INTR}}$ output. When the CPU writes a character into the transmit data holding register, these conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

The transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the transmit holding register, the TxD output remains in the marking (high) condition and the TxEMT/DSCHG status bit and the $\overline{\text{INTR}}$ output are asserted. Transmission resumes when the CPU loads a new character into the holding register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the send break command bit (CR3) high.

Table 2. CPU-Related Signals

PIN NAME	PIN NO.		INPUT/ OUTPUT	FUNCTION
	DIP	PLCC		
V_{CC}	21	26	I	+5V supply input
GND	3	4	I	Ground
RESET	18	21	I	A high on this input performs a master reset on the SCN2641. This signal asynchronously terminates any device activity and clears the mode, command and status registers. The device assumes the idle state and remains there until initialized with appropriate control words.
$A_1 - A_0$	9, 11	11, 13	I	Address lines used to select internal SCN2641 registers.
$\overline{R}$/W	12	14	I	Read command when low, write command when high.
$\overline{CE}$	10	12	I	Chip enable command. When low, indicates that control and data lines to the SCN2641 are valid and that the operation specified by the $\overline{R}$/W, A_1 and A_0 inputs should be performed. When high, places the $D_0 - D_7$ lines in the 3-State condition.
$D_7 - D_0$	7-4,1, 24-22	9,7-5, 2,1, 28,27	I/O	8-bit, 3-State data bus used to transfer commands, data and status between the SCN2641 and the CPU. D_0 is the least significant bit; D_7 the most significant bit.
$\overline{INTR}$	13	15	O	Interrupt request output (open drain). This output is asserted (low) under the following conditions. 1. When the transmitter holding register (THR) is ready to accept a data character from the CPU. This corresponds to assertion of status bit SR0. If this is the only condition asserting the output, the output will be negated (high) when the THR is loaded by the CPU, or if the transmitter is disabled via command register bit CR0. 2. When the receiver holding register (RHR) has a character ready to be read by the CPU. This corresponds to assertion of status bit SR1. If this is the only condition asserting the output, the output will be negated (high) when the RHR is read by the CPU, or if the receiver is disabled via command register bit CR2. 3. When the transmitter has completed serialization of the last character loaded by the CPU. This corresponds to assertion of status bit SR2. If this is the only condition asserting the output, the output will be negated (high) when the THR is loaded by the CPU. 4. When a change of state has occurred at the $\overline{DCD}$ input while either the receiver or the transmitter are enabled. This corresponds to assertion of status bit SR2. If this is the only condition asserting the output, the output will be negated (high) when the status register is read by the CPU.

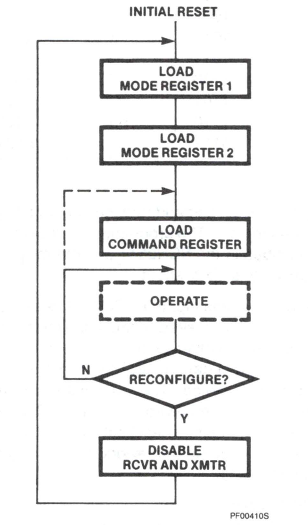
PROGRAMMING

Prior to initiating data communications, the SCN2641 operational mode must be programmed by performing write operations to the mode and command registers. The SCN2641 can be reconfigured at any time during program execution. A flowchart of the initialization process appears in Figure 1.

The internal registers of the SCN2641 are accessed by applying specific signals to the $\overline{CE}$, $\overline{R}$/W, A_1 and A_0 inputs. The conditions necessary to address each register are shown in Table 4.

Reading or loading the mode registers is done as follows: the first write (or read) operation addresses mode register 1 and a subsequent operation addresses mode register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointer is reset to mode register 1 by a RESET input or by performing a read command register operation, but is unaffected by any other read or write operation.

The SCN2641 register formats are summarized in Tables 5, 6, 7 and 8. Mode registers 1 and 2 define the general operational characteristics of the SCN2641, while the command register controls the operation within this basic framework. The SCN2641 indicates its status in the status register. These registers are cleared when a RESET input is applied.



NOTE:
Mode Register 1 must be written before 2 can be written. Mode register 2 need not be programmed if external clocks are used.

Figure 1. SCN2641 Initialization Flowchart

Mode Register 1 (MR1)

Table 5 illustrates mode register 1. Bits MR11 and MR10 select the baud rate multiplier. 1X, 16X and 64X multipliers are programmable if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7 or 8 bits. The character length does not include the parity bit (if programmed), and does not include the start and stop bits.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

MR17 and MR16 select character framing of 1, 1.5 or 2 stop bits (If 1X baud rate is programmed, 1.5 stop bits default to 1 stop bit on transmit).

The bits in the mode register affecting character assembly and disassembly (MR12 – MR16) can be changed dynamically (during active receive/transmit operation). The character mode register affects both the transmitter and receiver; therefore, character changes should be made when RxEN and TxEN = 0 or when TxEN = 1 and the transmitter is marking in half-duplex mode (RxEN = 0).

To effect assembly/disassembly of the next received/transmitted character, MR12 – MR15 must be changed within n-bit times of the assertion of RxRDY/TxRDY. (n = smaller of the new and old character lengths.)

Mode Register 2 (MR2)

Table 6 illustrates mode register 2. MR23, MR22, MR21 and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable as per Table 1. MR23 – MR20 are don't cares if external clocks are selected (MR25 – MR24 = 0). The individual rates are given in Table 1.

MR24 – MR27 select the receive and transmit clock source (either the BRG or an external input) and the function at pins 8 and 20 (Refer to Table 6).

Command Register (CR)

Table 7 illustrates the command register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. If the transmitter is disabled, it will complete the transmission of the character in the transmit shift register (if any) prior to terminating operation. The TxD output will then remain in the marking state (high), while the TxRDY and TxEMT status bits go low. Disabling the receiver causes the RxRDY status bit to go low. If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be negated. A 0-to-1 transition of CR2 will initiate start bit search on the second $\overline{\text{RxC}}$ rising edge following the transition.

Bit CR5 (RTS) controls the RTS output. Data at the output is the logical complement of the register data.

Setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The user should wait at least one bit time after terminating the break before loading the THR with the next character to be transmitted.

Setting CR4 causes the error flags in the status register (SR3, SR4 and SR5) to be cleared. This is a one-time command. There is no internal latch for this bit.

Table 3. Device-Related Signals

PIN NAME	DIP	PLCC	INPUT/ OUTPUT	FUNCTION
BRCLK	17	20	I	Clock input to the internal baud rate generator (see Table 1). Not required if external receiver and transmitter clocks are used.
$\overline{\text{RxC}}$/ BKDET	20	25	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. Data are sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin can be a 1X/16X clock or a break detect output pin.
RxD	2	3	I	Serial data input to the receiver. "Mark" is high, "space" is low.
$\overline{\text{TxC}}$	8	10	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, this pin can be a 1X/16X clock output.
TxD	16	19	O	Serial data output from the transmitter. "Mark" is high, "space" is low. Held in mark condition when the transmitter is disabled.
$\overline{\text{DCD}}$	14	16	I	Data carrier detect input. Must be low in order for the receiver to operate. Its complement appears as status register bit SR6. Causes a low output on $\overline{\text{INTR}}$ when its state changes if CR2 or CR0 = 1. If DCD goes high while receiving, the RxC is internally inhibited. Operation of the receiver resumes on the second $\overline{\text{RxC}}$ rising edge following assertion of DCD.
$\overline{\text{CTS}}$	15	17	I	Clear to send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the transmit shift register will be transmitted before termination.
$\overline{\text{RTS}}$	19	23	O	General-purpose output which is the complement of command register bit CR5. Normally used to indicate request to send. See Command Register (CR5) for details.

Table 4. Register Addressing

$\overline{\text{CE}}$	A_1	A_0	$\overline{\text{R}}$/W	FUNCTION
1	X	X	X	3-State data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Invalid
0	1	0	0	Read mode registers ½
0	1	0	1	Write mode registers ½
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE:
See AC characteristics section for timing requirements.

When CR5 (RTS) is set, the RTS pin is forced low. A 1-to-0 transition of CR5 will cause $\overline{\text{RTS}}$ to go high (inactive) one TxC time after the last serial bit has been transmitted. If a 1-to-0 transition of CR5 occurs while data is being transmitted, $\overline{\text{RTS}}$ will remain low (active) until both the THR and the transmit shift register are empty and then go high one TxC time later.

The SCN2641 can operate in one of four submodes. The operational submode is determined by CR7 and CR6. CR7 – CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the mode and status register instructions.

CR7 – CR6 = 01 places the SCN2641 in the automatic echo mode. Clocked, regenerated received data are automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU-to-receiver communications continue normally, but the CPU-to-transmitter link is disabled. Only the first character of a break condition is echoed. The TxD output will go high until the next valid start is detected. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. The $\overline{\text{INTR}}$ pin will reflect only the data set change condition.
4. The TxEN command (CR0) is ignored.

Two diagnostic submodes can also be configured. In local loopback mode (CR7 – CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{\text{RTS}}$ is connected to $\overline{\text{CTS}}$.
3. The receiver is clocked by the transmitter clock.
4. The $\overline{\text{RTS}}$ and TxD outputs are held high.
5. The $\overline{\text{CTS}}$, $\overline{\text{DCD}}$ and RxD inputs are ignored.

Additional requirements to operate in the local loopback mode are that CR0 (TxEN), CR1 and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the SCN2641.

The second diagnostic mode is the remote loopback mode (CR7 – CR6 = 11). In this mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. No data is sent to the local CPU, but the error status conditions (PE, OE, FE) are set.
4. The $\overline{\text{INTR}}$ output is held high.
5. CR0 (TxEN) is ignored.
6. All other signals operate normally.

Status Register

The data contained in the status register (as shown in Table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the transmitter ready (TxRDY) status bit. It is valid only when the transmitter is enabled. If equal to 0, it indicates that the transmit holding register has been loaded by the CPU and the data has not been transferred to the transmit shift register. If set equal to 1, it indicates that the holding register is ready to accept data from the CPU. This bit is initially set when the transmitter is enabled by CR0, unless a character has previously been loaded into the holding register. It is not set when the automatic echo or remote loopback modes are programmed. When this bit is set, the $\overline{\text{INTR}}$ output pin is low, except in the automatic echo and remote loopback modes.

SR1, the receiver ready (RxRDY) status bit, indicates the condition of the receive data holding register. If set, it indicates that a character has been loaded into the holding register from the receive shift register and is ready to be read by the CPU. If equal to 0, there is no new character in the holding register. This bit is cleared when the CPU reads the receive data holding register or when the receiver is disabled by CR2. When set, the $\overline{\text{INTR}}$ output is low.

Table 5. Mode Register 1 (MR1)

MR17	MR16	MR15	MR14	MR13	MR12	MR11	MR10
Stop Bit Length		**Parity Type**	**Parity Control**	**Character Length**		**Mode and Baud Rate Factor**	
00 = Invalid 01 = 1 stop bit 10 = 1½ stop bits 11 = 2 stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits		00 = Invalid 01 = 1X rate 10 = 16X rate 11 = 64X rate	

NOTE:
Baud rate factor applies only if external clock is selected. Factor is 16X if internal clock is selected.

Table 6. Mode Register 2 (MR2)

MR27 – MR24										MR23 – MR20
	TxC	RxC	Pin 8	Pin 20		TxC	RxC	Pin 8	Pin 20	Baud Rate Selection
0000	E	E	TxC	RxC	1000	E	E	NF	RxC/TxC	See baud rates in Table 1
0001	E	I	TxC	1X	1001	E	I	TxC	BKDET	
0010	I	E	1X	RxC	1010	I	E	NF	RxC	
0011	I	I	1X	1X	1011	I	I	1X	BKDET	
0100	E	E	TxC	RxC	1100	E	E	NF	RxC/TxC	
0101	E	I	TxC	16X	1101	E	I	TxC	BKDET	
0110	I	E	16X	RxC	1110	I	E	NF	RxC	
0111	I	I	16X	16X	1111	I	I	16X	BKDET	

NOTES:
E = External clock
I = Internal clock (BRG)
NF = No function; output not valid
1X and 16X are clock outputs

Table 7. Command Register (CR)

CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode		**Request To Send**	**Reset Error**	**Force Break**	**Receive Control (RxEN)**		**Transmit Control (TxEN)**
00 = Normal operation 01 = Automatic echo mode 10 = Local loopback 11 = Remote loopback		0 = Force $\overline{RTS}$ output high after TxSR serialization 1 = Force $\overline{RTS}$ output low	0 = Normal 1 = Reset error flags in status register (FE, OE, PE)	0 = Normal 1 = Force break	0 = Disable 1 = Enable	Not used. Must be programmed to '1'	0 = Disable 1 = Enable

Table 8. Status Register (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
	Data Carrier Detect	**Framing Error**	**Overrun Error**	**Parity Error**	**TxEMT/DSCHG**	**RxRDY**	**TxRDY**
Not used	0 = $\overline{DCD}$ input is high 1 = $\overline{DCD}$ input is low	0 = Normal 1 = Framing Error	0 = Normal 1 = Overrun Error	0 =Normal 1 =Parity error	0 = Normal 1 = Change in $\overline{DCD}$ or transmit shift register is empty	0 = Receive holding register empty 1 = Receive holding register has data	0 = Transmit holding register busy 1 = Transmit holding register empty

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{DCD}$ input (when CR2 or CR0 = 1) or that the transmit shift register has completed transmission of a character and no new character has been loaded into the transmit data holding register. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the transmit data holding register. The DSCHG condition is enabled when the TxEN = 1 or RxEN = 1. It is cleared when the status register is read by the CPU. If the status register is read twice and SR2 = 1 while SR6 remains unchanged, then a TxEMT condition exists. When SR2 is set, the $\overline{INTR}$ output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. This bit is cleared when the receiver is disabled and by a reset error command, CR4.

The overrun error status bit, SR4, indicates that the previous character loaded into the receive holding register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled and by the reset error command, CR4.

Bit SR5 signifies that the received character was not framed by a stop bit; i.e., only the first stop bit is checked. If RHR = 0 when SR5 = 1, a break condition is present. The bit is reset when the receiver is disabled and when the reset error command is given.

SR6 reflects the condition of the DCD input. A low input sets the status bit and a high input clears it.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Input voltage						
V_{IL} V_{IH}	Low High		 2		0.8	V
Output voltage						
V_{OL} V_{OH}[7]	Low High	$I_{OL} = 2.2mA$ $I_{OH} = -400\mu A$	 2.4		0.4	V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}			10	μA
3-State output leakage current						
I_{LH} I_{LL}	Data bus high Data bus low	$V_O = 4V$ $V_O = 0.45V$			10 10	μA
I_{CC}	Power supply current				150	mA

CAPACITANCE $T_A = 25°C$, $V_{CC} = 0V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Capacitance						
C_{IN} C_{OUT} $C_{I/O}$	Input Output Input/Output	fc = 1MHz Unmeasured pins tied to ground			20 20 20	pF

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Pulse width						
t_{RES}	Reset		1000			ns
t_{CE}	Chip enable		250			
t_{CED}	CE to CE delay		600			
Setup and hold time						
t_{AS}	Address setup		10			ns
t_{AH}	Address hold		10			
t_{CS}	$\overline{R}$/W control setup		10			
t_{CH}	$\overline{R}$/W control hold		10			
t_{DS}	Data setup for write		150			
t_{DH}	Data hold for write		10			
t_{RXS}	Rx data setup		300			
t_{RXH}	Rx data hold		350			
t_{DD}	Data delay time for read	$C_L = 150pF$			200	ns
t_{DF}	Data bus floating time for read	$C_L = 150pF$			100	
Input clock frequency						
f_{BRG}	Baud rate generator		1	3.6864	4	MHz
$f_{R/T}$[10]	$\overline{TxC}$ or $\overline{RxC}$		DC		1	
Clock state						
t_{BRH}[9]	Baud rate high		90			ns
t_{BRL}[9]	Baud rate low		90			
$t_{R/TH}$	$\overline{TxC}$ or $\overline{RxC}$ high		480			
$t_{R/TL}$[10]	$\overline{TxC}$ or $\overline{RxC}$ low		480			
t_{TXD}	TxD delay from falling edge of $\overline{TxC}$	$C_L = 150pF$			650	ns
t_{TCS}	Skew between TxD changing and falling edge of $\overline{TxC}$ output[8]	$C_L = 150pF$		0		

NOTES:

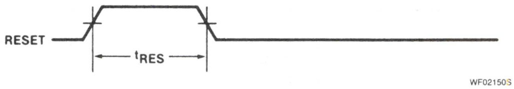
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. All time measurements are at the 50% level for inputs (except t_{BRH} and t_{BRL}) and at 0.8V and 2.0V for outputs. Input levels swing between 0.4V and 2.4V, with a transition time of 20ns maximum.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. $\overline{INTR}$ output is open drain.
8. Parameter applies when internal transmitter clock is used.
9. t_{BRH} and t_{BRL} measured at V_{IH} and V_{IL} respectively.
10. In asynchronous local loopback mode, using 1X clock, the following parameters apply:

    $f_{R/T} = 0.83MHz$ max

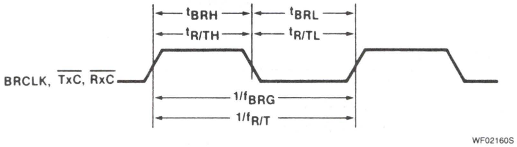
    $t_{R/TL} = 700ns$ min

Asynchronous Communications Interface SCN2641

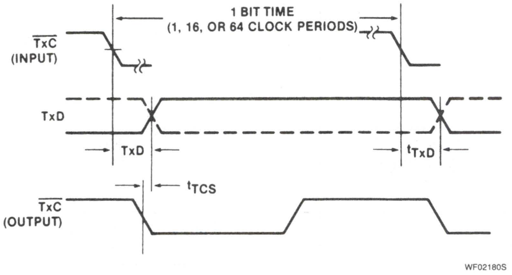
TIMING DIAGRAMS


RESET
t_{RES}
WF02150S

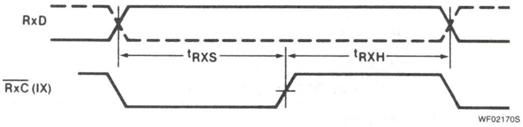
Reset


t_{BRH}
t_{BRL}
$t_{R/TH}$
$t_{R/TL}$
BRCLK, $\overline{TxC}$, $\overline{RxC}$
$1/f_{BRG}$
$1/f_{R/T}$
WF02160S

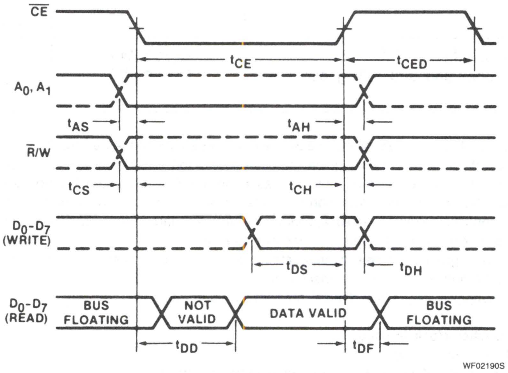
Clock


1 BIT TIME
(1, 16, OR 64 CLOCK PERIODS)
$\overline{TxC}$ (INPUT)
TxD
TxD
t_{TxD}
t_{TCS}
$\overline{TxC}$ (OUTPUT)
WF02180S

Transmit

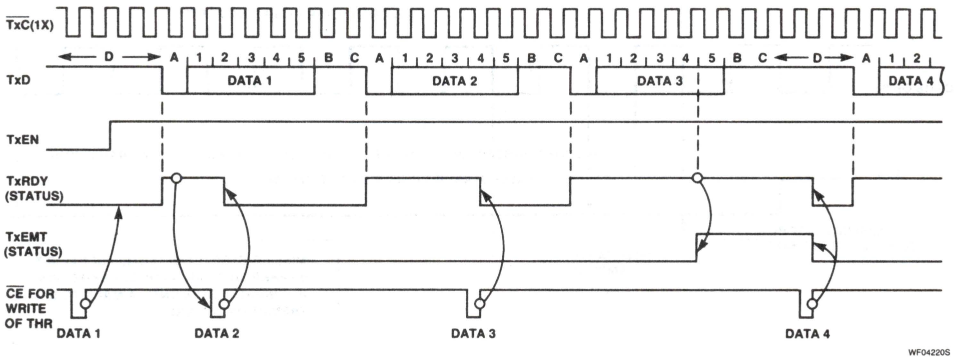

RxD
t_{RXS}
t_{RXH}
$\overline{RxC}$ (IX)
WF02170S

Receive


$\overline{CE}$
t_{CE}
t_{CED}
A_0, A_1
t_{AS}
t_{AH}
$\overline{R}$/W
t_{CS}
t_{CH}
D_0-D_7 (WRITE)
t_{DS}
t_{DH}
D_0-D_7 (READ)
BUS FLOATING
NOT VALID
DATA VALID
BUS FLOATING
t_{DD}
t_{DF}
WF02190S

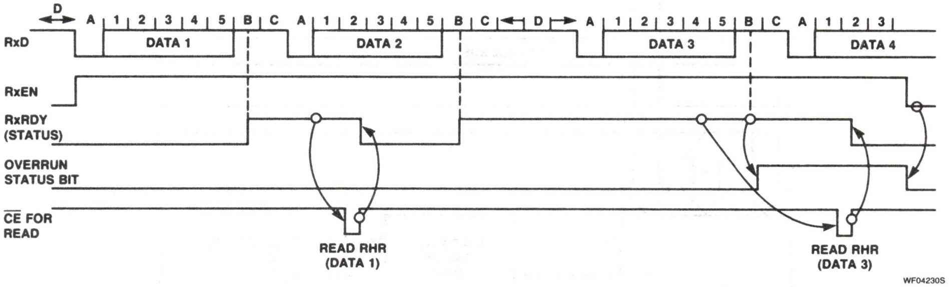
Read and Write

TIMING DIAGRAMS (Continued)


TxC(1X)
D
A
1
2
3
4
5
B
C
TxD
DATA 1
DATA 2
DATA 3
DATA 4
TxEN
TxRDY (STATUS)
TxEMT (STATUS)
CE FOR WRITE OF THR
DATA 1
DATA 2
DATA 3
DATA 4
WF04220S

NOTES:
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit.

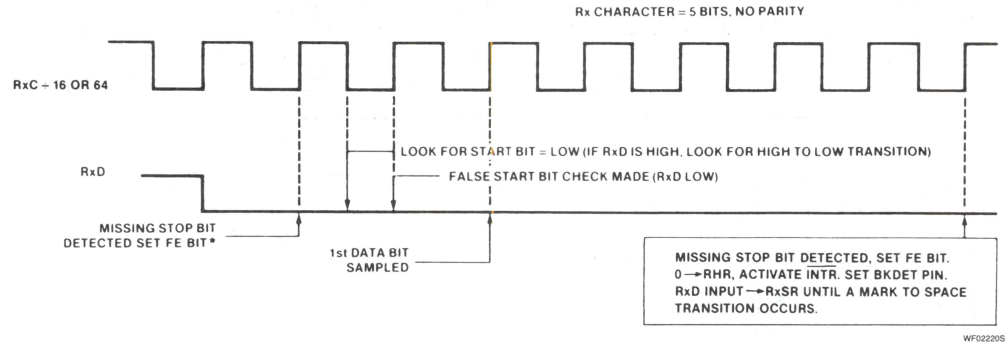
TxRDY, TxEMT (Shown for 5-bit characters, no parity, 2 stop bits)


D
A
1
2
3
4
5
B
C
RxD
DATA 1
DATA 2
DATA 3
DATA 4
RxEN
RxRDY (STATUS)
OVERRUN STATUS BIT
CE FOR READ
READ RHR (DATA 1)
READ RHR (DATA 3)
WF04230S

NOTES:
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
Only one stop bit is detected.

RxRDY (Shown for 5-bit characters, no parity, 2 stop bits)

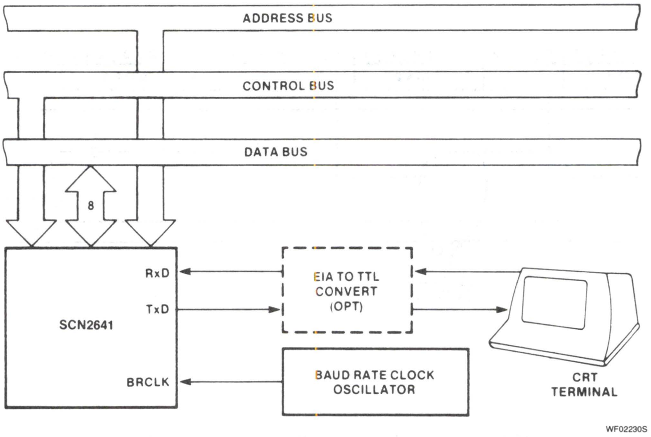
TIMING DIAGRAMS (Continued)


Rx CHARACTER = 5 BITS, NO PARITY
RxC ÷ 16 OR 64
RxD
LOOK FOR START BIT = LOW (IF RxD IS HIGH, LOOK FOR HIGH TO LOW TRANSITION)
FALSE START BIT CHECK MADE (RxD LOW)
MISSING STOP BIT DETECTED SET FE BIT*
1st DATA BIT SAMPLED
MISSING STOP BIT DETECTED, SET FE BIT.
0→RHR, ACTIVATE INTR. SET BKDET PIN.
RxD INPUT→RxSR UNTIL A MARK TO SPACE TRANSITION OCCURS.
WF02220S

NOTE:
*If the stop bit is present, the start bit search will commence immediately.

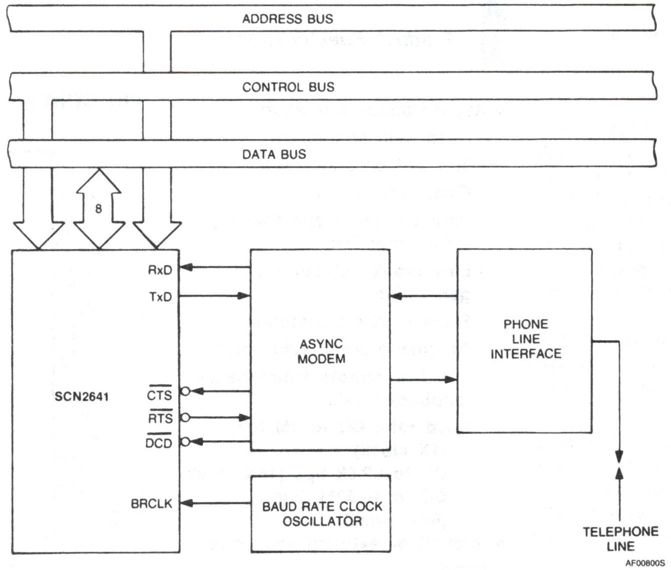
Break Detection Timing

TYPICAL APPLICATIONS


ADDRESS BUS
CONTROL BUS
DATA BUS
8
SCN2641
RxD
TxD
BRCLK
EIA TO TTL CONVERT (OPT)
BAUD RATE CLOCK OSCILLATOR
CRT TERMINAL
WF02230S

Asynchronous Interface to CRT Terminal

TYPICAL APPLICATIONS (Continued)


ADDRESS BUS
CONTROL BUS
DATA BUS
8
SCN2641
RxD
TxD
$\overline{CTS}$
$\overline{RTS}$
$\overline{DCD}$
BRCLK
ASYNC MODEM
PHONE LINE INTERFACE
BAUD RATE CLOCK OSCILLATOR
TELEPHONE LINE
AF00800S

Asynchronous Interface to Telephone Lines

Signetics

Microprocessor Products

SCN2651
Programmable Communications Interface (PCI)

Product Specification

DESCRIPTION

The Signetics SCN2651 PCI is a universal synchronous/asynchronous data communications controller chip designed for microcomputer systems. It interfaces directly to the Signetics SCN2650 microprocessor and may be used in a polled or interrupt driven system environment. The SCN2651 accepts programmed instructions from the microprocessor and supports many serial data communication disciplines, synchronous and asynchronous, in the full or half-duplex mode.

The PCI serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The SCN2651 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode.

The PCI is constructed using Signetics n-channel silicon gate depletion load technology and is packaged in a 28-pin DIP.

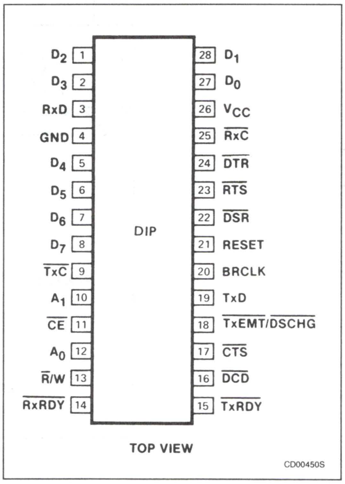
FEATURES

- **Synchronous operation**
  - **5- to 8-bit characters**
  - **Single or double SYN operation**
  - **Internal character synchronization**
  - **Transparent or non-transparent mode**
  - **Automatic SYN or DLE-SYN insertion**
  - **SYN or DLE stripping**
  - **Odd, even, or no parity**
  - **Local or remote maintenance loopback mode**
  - **Baud rate: DC to 1M bps (1X clock)**
- **Asynchronous operation**
  - **5- to 8-bit characters**
  - **1, 1 1/2 or 2 stop bits**
  - **Odd, even, or no parity**
  - **Parity, overrun and framing error detection**
  - **Line break detection and generation**
  - **False start bit detection**
  - **Automatic serial echo mode**
  - **Local or remote maintenance loopback mode**
  - **Baud rate: DC to 1M bps (1X clock) DC to 62.5k bps (16X clock) DC to 15.625k bps (64X clock)**
- **Internal or external baud rate clock**
- **16 Internal rates - 50 to 19,200 baud**
- **Double buffered transmitter and receiver**
- **Full or half duplex operation**
- **TTL compatible inputs and outputs**
- **Single 5V power supply**
- **No system clock required**
- **28-pin dual in-line package**

APPLICATIONS

- **Intelligent terminals**
- **Network processors**
- **Front-end processors**
- **Remote data concentrators**
- **Computer to computer links**
- **Serial peripherals**

PIN CONFIGURATION

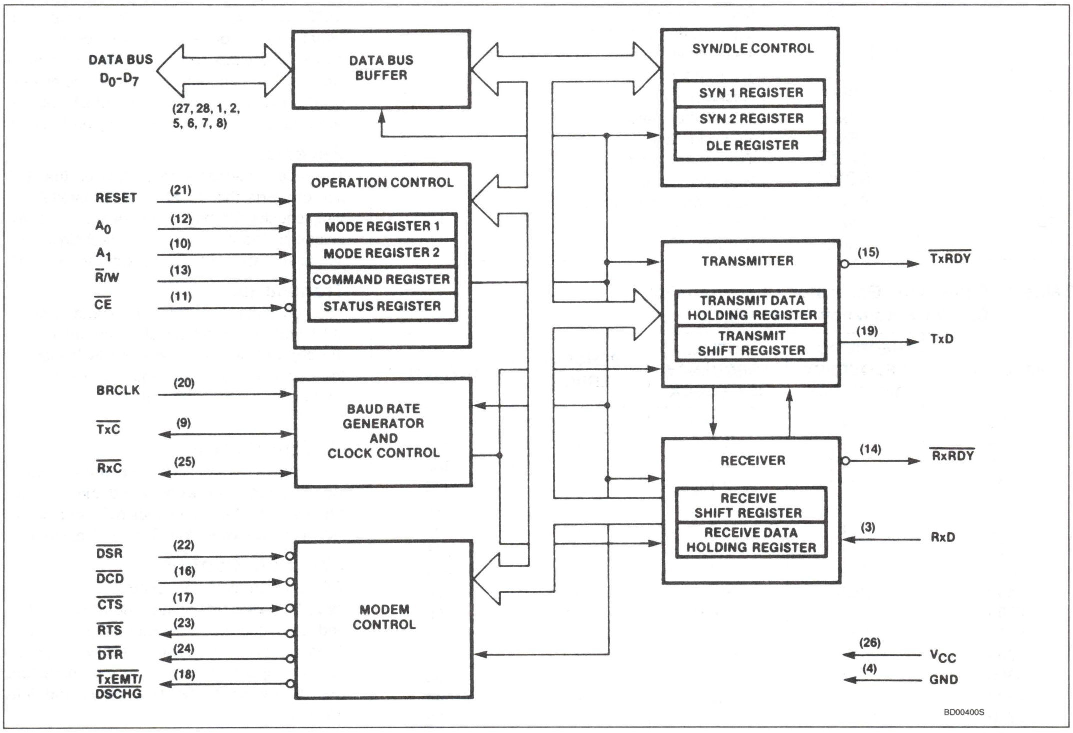

D_2 1
D_3 2
RxD 3
GND 4
D_4 5
D_5 6
D_6 7
D_7 8
$\overline{TxC}$ 9
A_1 10
$\overline{CE}$ 11
A_0 12
$\overline{R}/W$ 13
$\overline{RxRDY}$ 14
28 D_1
27 D_0
26 V_{CC}
25 $\overline{RxC}$
24 $\overline{DTR}$
23 $\overline{RTS}$
22 $\overline{DSR}$
21 RESET
20 BRCLK
19 TxD
18 $\overline{TxEMT}/\overline{DSCHG}$
17 $\overline{CTS}$
16 $\overline{DCD}$
15 $\overline{TxRDY}$
DIP
TOP VIEW
CD00450S

Programmable Communications Interface (PCI) SCN2651

ORDERING INFORMATION

PACKAGES	V_{CC} = 5V ±5%		
	Commercial	**Automotive**	**Military**
	0 to +70°C	**−40°C to + 85°C**	**−55°C to +125°C**
Ceramic DIP	SCN2651CC1I28	Contact Factory	Contact Factory
Plastic DIP	SCN2651CC1N28	Contact Factory	Not available

BLOCK DIAGRAM


DATA BUS
D_0-D_7
(27, 28, 1, 2, 5, 6, 7, 8)
DATA BUS BUFFER
SYN/DLE CONTROL
SYN 1 REGISTER
SYN 2 REGISTER
DLE REGISTER
RESET (21)
A_0 (12)
A_1 (10)
$R/\overline{W}$ (13)
$\overline{CE}$ (11)
OPERATION CONTROL
MODE REGISTER 1
MODE REGISTER 2
COMMAND REGISTER
STATUS REGISTER
TRANSMITTER
TRANSMIT DATA HOLDING REGISTER
TRANSMIT SHIFT REGISTER
(15) $\overline{TxRDY}$
(19) TxD
BRCLK (20)
$\overline{TxC}$ (9)
$\overline{RxC}$ (25)
BAUD RATE GENERATOR AND CLOCK CONTROL
RECEIVER
RECEIVE SHIFT REGISTER
RECEIVE DATA HOLDING REGISTER
(14) $\overline{RxRDY}$
(3) RxD
$\overline{DSR}$ (22)
$\overline{DCD}$ (16)
$\overline{CTS}$ (17)
$\overline{RTS}$ (23)
$\overline{DTR}$ (24)
$\overline{TxEMT}/\overline{DSCHG}$ (18)
MODEM CONTROL
(26) V_{CC}
(4) GND
BD00400S

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	TYPE
27, 28, 1, 2, 5 – 8	$D_0 - D_7$	8-bit data bus	I/O
21	RESET	Reset	I
12,10	$A_0 - A_1$	Internal register select lines	I
13	$\overline{\text{R}}$/W	Read or write command	I
11	$\overline{\text{CE}}$	Chip enable input	I
22	$\overline{\text{DSR}}$	Data set ready	I
24	$\overline{\text{DTR}}$	Data terminal ready	O
23	$\overline{\text{RTS}}$	Request to send	O
17	$\overline{\text{CTS}}$	Clear to send	I
16	$\overline{\text{DCD}}$	Data carrier detected	I
18	$\overline{\text{TxEMT/DSCHG}}$	Transmitter empty or data set change	O
9	$\overline{\text{TxC}}$	Transmitter clock	I/O
25	$\overline{\text{RxC}}$	Receiver clock	I/O
19	TxD	Transmitter data	O
3	RxD	Receiver data	I
15	$\overline{\text{TxRDY}}$	Transmitter ready	O
14	$\overline{\text{RxRDY}}$	Receiver ready	O
20	BRCLK	Baud rate generator clock	I
26	V_{CC}	+5V supply	I
4	GND	Ground	I

Table 1. Baud Rate Generator Characteristics Crystal Frequency = 5.0688MHz

BAUD RATE	THEORETICAL FREQUENCY 16X CLOCK	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
50	0.8KHz	0.8KHz	—	6336
75	1.2	1.2	—	4224
110	1.76	1.76	—	2880
134.5	2.152	2.1523	0.016	2355
150	2.4	2.4	—	2112
300	4.8	4.8	—	1056
600	9.6	9.6	—	528
1200	19.2	19.2	—	264
1800	28.8	28.8	—	176
2000	32.0	32.081	0.253	158
2400	38.4	38.4	—	132
3600	57.6	57.6	—	88
4800	76.8	76.8	—	66
7200	115.2	115.2	—	44
9600	153.6	153.6	—	33
19200*	307.2	316.8	3.125	16

NOTES:

1. *Error at 19200 can be reduced to zero by using crystal frequency 4.9152MHz.
2. 16X clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1X.

BLOCK DIAGRAM

The PCI consists of these six major sections: the transmitter, receiver, timing, operation control, modem control and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers 1 and 2, the command register, and the status register. Details of register addressing and protocol are presented in the PCI programming section of this data sheet.

Timing

The PCI contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full duplex operation (see Table 1).

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

INTERFACE SIGNALS

The PCI interface signals can be grouped into two types: the CPU-related signals (shown in Table 2), which interface the SCN2651 to the microprocessor system, and the device-related signals (shown in Table 3), which are used to interface to the communications device or system.

Table 2. CPU-Related Signals

PIN NAME	PIN NO.	INPUT/OUTPUT	FUNCTION
V_{CC}	26	I	+5V supply input
GND	4	I	Ground
RESET	21	I	A high on this input performs a master reset on the SCN2651. This signal asynchronously terminates any device activity and clears the mode, command and status registers. The device assumes the idle state and remains there until initialized with the appropriate control words.
$A_1 - A_0$	10, 12	I	Address lines used to select internal PCI registers.
$\overline{\text{R}}$/W	13	I	Read command when low, write command when high.
$\overline{\text{CE}}$	11	I	Chip enable command. When low, indicates that control and data lines to the PCI are valid and that the operation specified by the $\overline{\text{R}}$/W, A_1 and A_0 inputs should be performed. When high, places the $D_0 - D_7$ lines in the 3-State condition.
$D_7 - D_0$	8, 7, 6, 5, 2, 1, 28, 27	I/O	8-bit, 3-State data bus used to transfer commands, data and status between PCI and the CPU. D_0 is the least significant bit; D_7 the most significant bit.
$\overline{\text{TxRDY}}$	15	O	This output is the complement of status register bit SR0. When low, it indicates that the transmit data holding register (THR) is ready to accept a data character from the CPU. It goes high when the data character is loaded. This output is valid only when the transmitter is enabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{\text{RxRDY}}$	14	O	This output is the complement of status register bit SR1. When low, it indicates that the receive data holding register (RHR) has a character ready for input to the CPU. It goes high when the RHR is read by the CPU, and also when the receiver is disabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$	18	O	This output is the complement of status register bit SR2. When low, it indicates that the transmitter has completed serialization of the last character loaded by the CPU, or that a change of state of the $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$ inputs has occurred. This output goes high when the status register is read by the CPU, if the TxEMT condition does not exist. Otherwise, the THR must be loaded by the CPU for this line to go high. It is an open drain output which can be used as an interrupt to the CPU.

OPERATION

The functional operation of the SCN2651 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the PCI programming section of this data sheet.

After programming, the PCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The SCN2651 is conditioned to receive data when the $\overline{\text{DCD}}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high to low transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and the stop bit(s) have been assembled. The data is then transferred to the receive data holding register, the RxRDY bit in the status register is set, and the $\overline{\text{RxRDY}}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the holding register are set to zero. The parity error, framing error, and overrun error status bits are strobed into the status register on the positive going edge of $\overline{\text{RxC}}$ corresponding to the received character boundary. If a break condition is detected (RxD is low for the entire character as well as the stop bit[s]), only one character consisting of all zeros (with the FE status bit set) will be transferred to the holding register. The RxD input must return to a high condition before a search for the next start bit begins.

When the PCI is initialized into the synchronous mode, the receiver first enters the hunt mode on a 0 to 1 transition of RxEN (CR2). In this mode, as data is shifted into the receiver shift register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN detect status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN detect bit to be set. Otherwise, the PCI returns to the hunt mode. (Note that the sequence SYN1 – SYN1 – SYN2 will not achieve synchronization). When synchronization has been achieved, the PCI continues to assemble characters and transfer them to the holding register, setting the RxRDY status bit and asserting the $\overline{\text{RxRDY}}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN detect status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the Holding Register. Note that the SYN characters used to establish initial synchronization are not transferred to the holding register in any case.

Table 3. Device-Related Signals

PIN NAME	PIN NO.	INPUT/OUTPUT	FUNCTION
BRCLK	20	I	5.0688MHz clock input to the internal baud rate generator. Not required if external receiver and transmitter clocks are used.
$\overline{\text{RxC}}$	25	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. Data is sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin becomes an output at 1X the programmed baud rate.*
$\overline{\text{TxC}}$	9	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, the pin becomes an output at 1X the programmed baud rate.*
RxD	3	I	Serial data input to the receiver. "Mark" is high, "Space" is low.
TxD	19	O	Serial data output from the transmitter. "Mark" is high, "space" is low. Held in mark condition when the transmitter is disabled.
$\overline{\text{DSR}}$	22	I	General purpose input which can be used for data set ready or ring indicator condition. Its complement appears as status register bit SR7. Causes a low output on $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ when its state changes.
$\overline{\text{DCD}}$	16	I	Data carrier detect input. Must be low in order for the receiver to operate. Its complement appears as status register bit SR6. Causes a low output on $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ when its state changes.
$\overline{\text{CTS}}$	17	I	Clear to send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the transmit shift register will be transmitted before termination.
$\overline{\text{DTR}}$	24	O	General purpose output which is the complement of command register bit CR1. Normally used to indicate data terminal ready.
$\overline{\text{RTS}}$	23	O	General purpose output which is the complement of command register bit CR5. Normally used to indicate request to send.

NOTE:
*$\overline{\text{RxC}}$ and $\overline{\text{TxC}}$ outputs have short circuit protection max. $C_L = 100\text{pF}$

Transmitter

The PCI is conditioned to transmit data when the $\overline{\text{CTS}}$ input is low and the TxEN command register bit is set. The SCN2651 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{\text{TxRDY}}$ output. When the CPU writes a character into the transmit data holding register, these conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the transmit holding register, the TxD output remains in the marking (high) condition and the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the holding register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the send break command bit high.

In the synchronous mode, when the SCN2651 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the PCI unless the CPU fails to send a new character to the PCI by the time the transmitter has completed sending the previous character.

Since synchronous communication does not allow gaps between characters, the PCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1 – SYN2 doublets, or DLE – SYN1 doublets, depending on the state of MR16 and MR17. Normal transmission of the message resumes when a new character is available in the transmit data holding register. If the send DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character in THR.

PCI PROGRAMMING

Prior to initiating data communications, the SCN2651 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The PCI can be reconfigured at any time during program execution. However, if the change has an effect on the reception of a character the receiver should be disabled. Alternatively if the change is made 1½ RxC periods after RxRDY goes active it will affect the next character assembly. A flowchart of the initialization process appears in Figure 1.

The internal registers of the PCI are accessed by applying specific signals to the $\overline{\text{CE}}$, $\overline{\text{R}}$/W, A_1 and A_0 inputs. The conditions necessary to address each register are shown in Table 4.

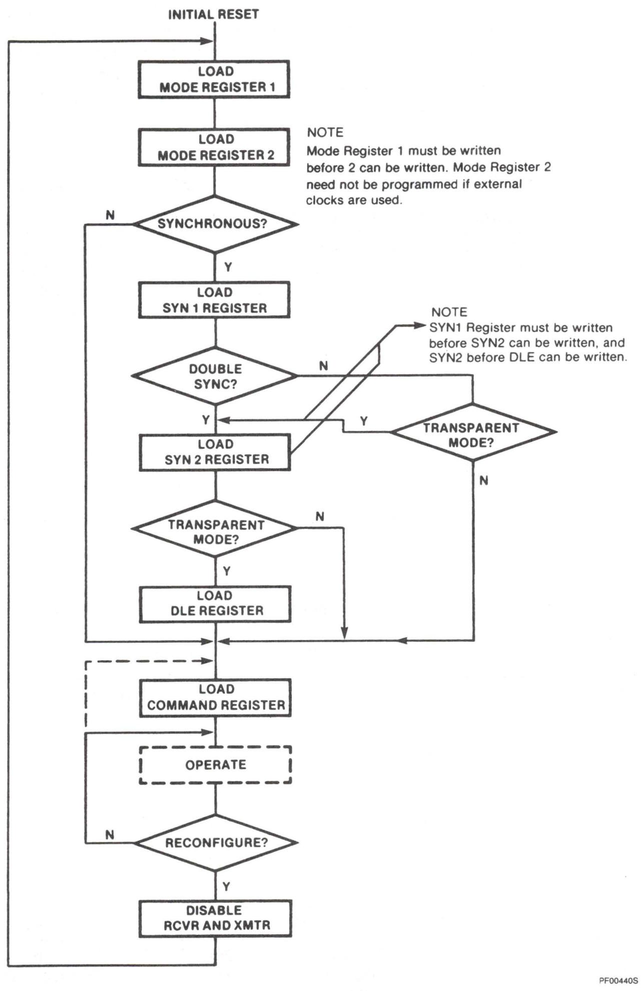

INITIAL RESET
LOAD MODE REGISTER 1
LOAD MODE REGISTER 2
NOTE
Mode Register 1 must be written before 2 can be written. Mode Register 2 need not be programmed if external clocks are used.
N
SYNCHRONOUS?
Y
LOAD SYN 1 REGISTER
NOTE
SYN1 Register must be written before SYN2 can be written, and SYN2 before DLE can be written.
DOUBLE SYNC?
N
Y
Y
TRANSPARENT MODE?
LOAD SYN 2 REGISTER
N
TRANSPARENT MODE?
N
Y
LOAD DLE REGISTER
LOAD COMMAND REGISTER
OPERATE
N
RECONFIGURE?
Y
DISABLE RCVR AND XMTR
PF00440S

Figure 1. SCN2651 Initialization Flowchart

Table 4. SCN2651 Register Addressing

$\overline{CE}$	A_1	A_0	$\overline{R}/W$	FUNCTION
1	X	X	X	3-State data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Write SYN1/SYN2/DLE registers
0	1	0	0	Read mode registers ½
0	1	0	1	Write mode registers ½
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE:
See AC Characteristics section for timing requirements.

The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions $A_1 = 0$, $A_0 = 1$, and $\overline{R}/W = 1$. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses mode register 1, and a subsequent operation addresses mode register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 register and mode register 1 by a RESET input or by performing a "read command register" operation, but are unaffected by any other read or write operation.

The SCN2651 register formats are summarized in tables 5, 6, 7 and 8. Mode registers 1 and 2 define the general operational characteristics of the PCI, while the command register controls the operation within this basic framework. The PCI indicates its status in the status register. These registers are cleared when a RESET input is applied.

Mode Register 1 (MR1)

Table 5 illustrates mode register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1X multiplier. 1X, 16X, and 64X multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7, or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asynchronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits. (If 1X baud rate is programmed, 1.5 stop bits default to 1 stop bit on transmit). In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1 – SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE – SYN1 is used for character fill and SYN detect, but the normal synchronization sequence is used. Also DLE stripping and DLE detect (with MR14 = 0) are enabled.

Mode Register 2 (MR2)

Table 6 illustrates mode register 2. MR23, MR22, MR21, and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable. When driven by a 5.0688MHz input at the BRCLK input (pin 20), the BRG output has zero error except at 134.5, 2000, and 19,200 baud, which have errors of +0.016%, +0.235%, and +3.125% respectively.

TR25 and MR24 select either the BRG or the external inputs $\overline{\text{TxC}}$ and $\overline{\text{RxC}}$ as the clock source for the transmitter and receiver, respectively. If the BRG clock is selected, the baud rate factor in asynchronous mode is 16X regardless of the factor selected by MR11 and MR10. In addition, the corresponding clock pin provides an output at 1X the baud rate.

Table 5. Mode Register 1 (MR1)

MR17	MR16	MR15	MR14	MR13	MR12	MR11	MR10
		Parity Type	**Parity Control**	**Character Length**		**Mode and Baud Rate Factor**	
Asynch: Stop bit length 00 = Invalid 01 = 1 Stop bit 10 = 1½ Stop bits 11 = 2 Stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 Bits 01 = 6 Bits 10 = 7 Bits 11 = 8 Bits		00 = Synchronous 1X rate 01 = Asynchronous 1X rate 10 = Asynchronous 16X rate 11 = Asynchronous 64X rate	
Synch: Number of syn char 0 = Double SYN 1 = Single SYN	**Synch: Transparency control** 0 = Normal 1 = Transparent						

NOTE:
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected. Mode must be selected (MR11, MR10) in any case.

Table 6. Mode Register 2 (MR2)

MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20
		Transmitter Clock	**Receiver Clock**	**Baud Rate Selection**			
Not used		0 = External 1 = Internal	0 = External 1 = Internal	0000 = 50 Baud 0001 = 75 0010 = 110 0011 = 134.5 0100 = 150 0101 = 300 0110 = 600 0111 = 1200		1000 = 1800 Baud 1001 = 2000 1010 = 2400 1011 = 3600 1100 = 4800 1101 = 7200 1110 = 9600 1111 = 19,200	

Table 7. Command Register (CR)

CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode		**Request to Send**	**Reset Error**		**Receive Control (RxEN)**	**Data Terminal Ready**	**Transmit Control (TxEN)**
00 = Normal operation 01 = Asynch: automatic echo mode Synch: SYN and/or DLE stripping mode 10 = Local Loopback 11 = Remote Loopback		0 = Force $\overline{\text{RTS}}$ output high 1 = Force $\overline{\text{RTS}}$ output low	0 = Normal 1 = Reset error flag in status reg (FE, OE, PE/DLE DETECT)	**Asynch: Force break** 0 = Normal 1 = Force break	0 = Disable 1 = Enable	0 = Force $\overline{\text{DTR}}$ output high 1 = Force $\overline{\text{DTR}}$ output low	0 = Disable 1 = Enable
				Synch Send DLE 0 = Normal 1 = Send DLE			

Table 8. Status Register (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	**Data Carrier Detect**	**FE/SYN Detect**	**Overrun**	**PE/DLE Detect**	**TxEMT/DSCHG**	**RxRDY**	**TxRDY**
0 = $\overline{\text{DSR}}$ Input is high 1 = $\overline{\text{DSR}}$ Input is low	0 = $\overline{\text{DCD}}$ Input is high 1 = $\overline{\text{DCD}}$ Input is low	**Asynch:** 0 = Normal 1 = Framing ERROR **Synch:** 0 = Normal 1 = SYN char detected	0 = Normal 1 = Overrun error	**Asynch:** 0 = Normal 1 = Parity error **Synch:** 0 = Normal 1 = Parity error or DLE char received	0 = Normal 1 = Change in $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$, or transmit shift register is empty	0 = Receive holding reg empty 1 = Receive holding reg has data	0 = Transmit holding reg busy 1 = Transmit holding reg empty

Command Register (CR)

Table 7 illustrates command register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. A 0 to 1 transition of CR2 forces start bit search (async mode) or hunt mode (sync mode) on the second $\overline{\text{RxC}}$ rising edge. Disabling the receiver causes $\overline{\text{RxRDY}}$ to go high (inactive). If the transmitter is disabled, it will complete the transmission of the character in the transmit shift register (if any) prior to terminating operation. The TxD output will then remain in the marking state (high) while the $\overline{\text{TxRDY}}$ and $\overline{\text{TxEMT}}$ will go high (inactive). If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be neglected.

Bits CR1 (DTR) and CR5 (RTS) control the $\overline{\text{DTR}}$ and $\overline{\text{RTS}}$ outputs. Data at the outputs is the logical complement of the register data.

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The TxD line will go high for at least one bit time before beginning transmission of the next character in the transmit data holding register. In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the transmit data holding register. CR3 should be reset in response to the next TxRDY.

Setting CR4 causes the error flags in the status register (SR3, SR4, and SR5) to be cleared. This is a one time command. There is no internal latch for this bit.

The PCI can operate in one of four submodes within each major mode (synchronous or asynchronous). The operational submode is determined by CR7 and CR6. CR7 – CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the mode and status register instructions.

In asynchronous mode, CR7 – CR6 = 01 places the PCI in the automatic echo mode. Clocked, regenerated received data is automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU to receiver communications continues normally, but the CPU to transmitter link is disabled. Only the first character of a break condition is echoed. The TxD output will go high until the next valid start is detected.

The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. $\overline{\text{TxRDY}}$ output = 1.
4. The $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ pin will reflect only the data set change condition.
5. The TxEN command (CR0) is ignored.

In synchronous mode, CR7 – CR6 = 01 places the PCI in the automatic SYN/DLE stripping mode. The exact action taken depends on the setting of bits MR17 and MR16:

1. In the non-transparent, single SYN mode (MR17 – MR16 = 10), characters in the data stream matching SYN1 are not transferred to the receive data holding register (RHR).
2. In the non-transparent, double SYN mode (MR17 – MR16 = 00), characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR. However, only the first SYN1 of an SYN1 – SYN1 pair is stripped.
3. In transparent mode (MR16 = 1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR. However, only the first DLE of a DLE – DLE pair is stripped.

Note that automatic stripping mode does not affect the setting of the DLE detect and SYN detect status bits (SR3 and SR5).

Two diagnostic submodes can also be configured. In local loopback mode (CR7 – CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{\text{DTR}}$ is connected to $\overline{\text{DCD}}$ and $\overline{\text{RTS}}$ is connected to $\overline{\text{CTS}}$.
3. The receiver is clocked by the transmit clock.
4. The $\overline{\text{DTR}}$, $\overline{\text{RTS}}$ and $\overline{\text{TxD}}$ outputs are held high.
5. The $\overline{\text{CTS}}$, $\overline{\text{DCD}}$, $\overline{\text{DSR}}$ and RxD inputs are ignored.

Additional requirements to operate in the local loopback mode are that CR0 (TxEN), CR1 (DTR), and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the PCI.

The second diagnostic mode is the remote loopback mode (CR7 – CR6 = 11). In this mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. No data is sent to the local CPU, but the error status conditions (PE, OE, FE) are set.
4. The $\overline{\text{RxRDY}}$, $\overline{\text{TxRDY}}$, and $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ outputs are held high.
5. CR0 (TxEN) is ignored.
6. All other signals operate normally.

Status Register

The data contained in the status register (as shown in Table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the transmitter ready (TxRDY) status bit. It, and its corresponding output, are valid only when the transmitter is enabled. If equal to 0, it indicates that the transmit data holding register has been loaded by the CPU and the data has not been transferred to the transmit shift register. If set equal to 1, it indicates that the Holding Register is ready to accept data from the CPU. This bit is initially set when the transmitter is enabled by CR0, unless a character has previously been loaded into the holding register. It is not set when the automatic echo or remote loopback modes are programmed. When this bit is set, the $\overline{\mathrm{TxRDY}}$ output pin is low. In the automatic echo and remote loopback modes, the output is held high.

SR1, the receiver ready (RxRDY) status bit, indicates the condition of the receive data holding register. If set, it indicates that a character has been loaded into the holding register from the receive shift register and is ready to be read by the CPU. If equal to zero, there is no new character in the holding register. This bit is cleared when the CPU reads the receive data holding register or when the receiver is disabled by CR2. When set, the $\overline{\mathrm{RxRDY}}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{\mathrm{DSR}}$ or $\overline{\mathrm{DCD}}$ inputs or that the transmit shift register has completed transmission of a character and no new character has been loaded into the transmit data holding register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the transmit data holding register. The DSCHG condition is enabled when TxEN = 1 or RxEN = 1. If the status register is read twice and SR2 = 1 while SR6 and SR7 remain unchanged, then a TxEMT condition exists. It is cleared when the status register is read by the CPU. When SR2 is set, the TxEMT/DSCHG output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching the DLE register has been received. However, only the first DLE of two successive DLEs will set SR3. This bit is cleared when the receiver is disabled and by the reset error command, CR4.

The overrun error status bit, SR4, indicates that the previous character loaded into the receive holding register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled and by the reset error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by the programmed number of stop bits. (If 1.5 stop bits are programmed, only the first stop bit is checked.) If RHR = 0 when SR5 = 1 a break condition is present. In synchronous non-transparent mode (MR16 = 0), it indicates receipt of the SYN1 character is single SYN mode or the SYN1 - SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1 - SYN2) and, after synchronization has been achieved, when a DLE - SYN1 pair is received. The bit is reset when the receiver is disabled, when the reset error command is given in asynchronous mode, and when the status register is read by the CPU in the synchronous mode.

SR6 and SR7 reflect the conditions of the $\overline{\mathrm{DCD}}$ and $\overline{\mathrm{DSR}}$ inputs respectively. A low input sets its corresponding status bit and a high input clears it.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	Note 4	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +6	V

DC ELECTRICAL CHARACTERISTICS[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Input voltage						
V_{IL}	Low				0.8	V
V_{IH}	High		2			
Output voltage						
V_{OL}	Low	I_{OL} = 1.6mA			0.4	V
V_{OH}	High	I_{OH} = −100μA	2.4			
I_{IL}	Input leakage current	V_{IN} = 0 to 5.25V	−10		10	μA
3-State output leakage current						
I_{LH}	Data bus high	V_O = 4.0V	−10		10	μA
I_{LL}	Data bus low	V_O = 0.45V	−10		10	
I_{CC}	Power supply current				150	mA

CAPACITANCE $T_A = 25°C$, $V_{CC} = 0V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Capacitance						
C_{IN}	Input	fc = 1MHz			20	pF
C_{OUT}	Output	Unmeasured pins tied			20	
$C_{I/O}$	Input/Output	to ground			20	

Programmable Communications Interface (PCI) SCN2651

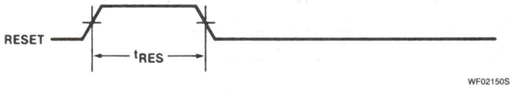
AC ELECTRICAL CHARACTERISTICS[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Pulse width						
t_{RES}	Reset		1000			ns
t_{CE}	Chip enable		300			
Setup and hold time						
t_{AS}	Address setup		20			ns
t_{AH}	Address hold		20			
t_{CS}	$\overline{R}$/W control setup		20			
t_{CH}	$\overline{R}$/W control hold		20			
t_{DS}	Data setup for write		225			
t_{DH}	Data hold for write		0			
t_{RXS}	Rx data setup		300			
t_{RXH}	Rx data hold		350			
t_{DD}	Data delay time for read	$C_L = 100pF$			250	ns
t_{DF}	Data bus floating time for read	$C_L = 100pF$			150	ns
t_{CED}	$\overline{CE}$ to $\overline{CE}$ delay		700			ns
Input clock frequency						
f_{BRG}	Baud rate generator		1.0	5.0688	5.0738	MHz
$f_{R/T}$[10]	$\overline{TxC}$ or $\overline{RxC}$		DC		1	
Clock width						
t_{BRH}[9]	Baud rate high		70			ns
t_{BRL}[9]	Baud rate low		70			
$t_{R/TH}$	$\overline{TxC}$ or $\overline{RxC}$ high		500			
$t_{R/TL}$[10]	$\overline{TxC}$ or $\overline{RxC}$ low		500			
t_{TXD}	MxD delay from falling edge of $\overline{TxC}$	$C_L = 100pF$			650	ns
t_{TCS}	Skew between TxD changing and falling edge of $\overline{TxC}$ output[8]	$C_L = 100pF$		0		ns

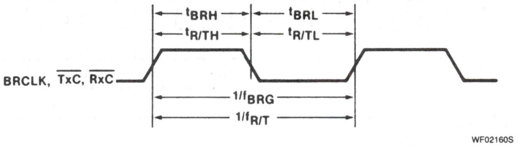
NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified. See ordering code table for applicable temperature range and operating supply range.
5. All voltage measurements are referenced to ground. All time measurements are at the 50% level for inputs (except t_{BRH} and T_{BRL}) and at 0.8V and 2V for outputs. Input levels for testing 0.45V and 2.4V.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. TxRDY, RxRDY and TxEMT/DSCHG outputs are open drain.
8. Parameter applies when internal transmitter clock is used.
9. Under test conditions of 5.0688MHz, f_{BRG}, t_{BRH}, and t_{BRL} measured at V_{IH} and V_{IL} respectively.
10. $t_{R/T}$ and $t_{R/TL}$ shown for all modes except local loopback. For local loopback mode $f_{R/T} = 0.7MHz$ and $t_{R/TL} = 700ns$ min.

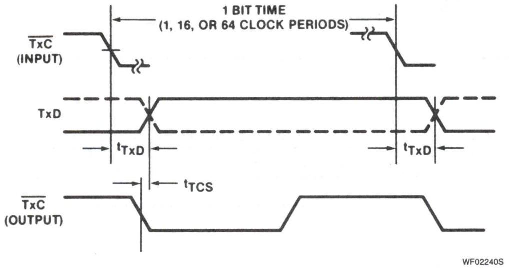
TIMING DIAGRAMS


RESET
tRES
WF02150S

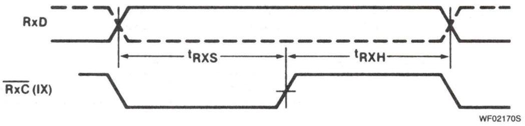
Reset


tBRH
tBRL
tR/TH
tR/TL
BRCLK, TxC, RxC
1/fBRG
1/fR/T
WF02160S

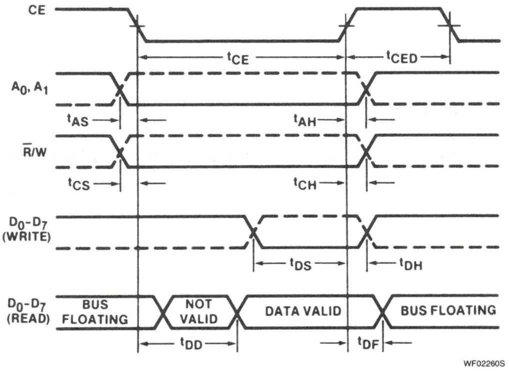
Clock


1 BIT TIME
(1, 16, OR 64 CLOCK PERIODS)
TxC
(INPUT)
TxD
tTxD
tTxD
tTCS
TxC
(OUTPUT)
WF02240S

Transmit

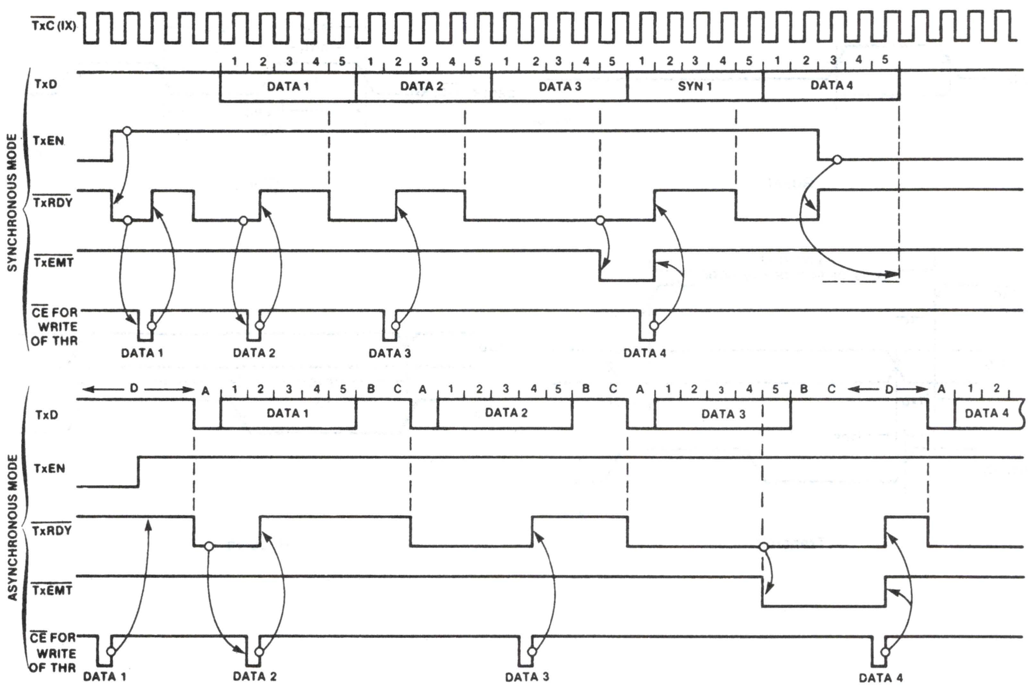

RxD
tRXS
tRXH
RxC (IX)
WF02170S

Receive


CE
tCE
tCED
A0, A1
tAS
tAH
R/W
tCS
tCH
D0-D7
(WRITE)
tDS
tDH
D0-D7
(READ)
BUS
FLOATING
NOT
VALID
DATA VALID
BUS FLOATING
tDD
tDF
WF02260S

Read and Write

TIMING DIAGRAMS (Continued)

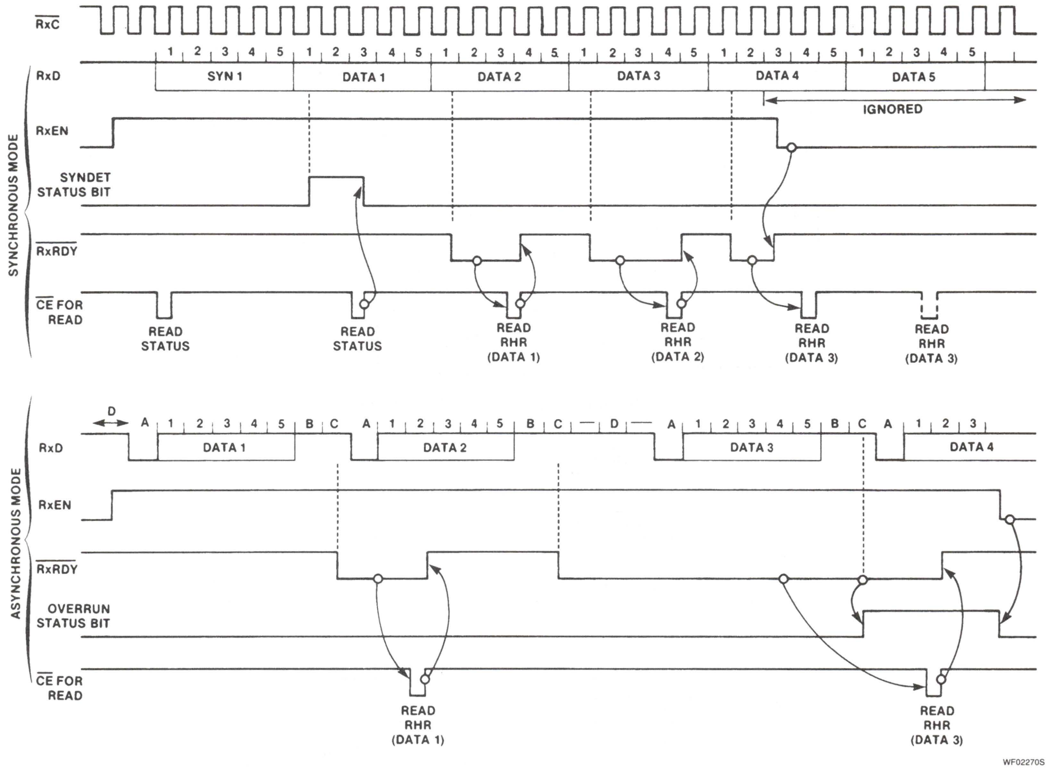

TxC (IX)
SYNCHRONOUS MODE
TxD
1 2 3 4 5
DATA 1
DATA 2
DATA 3
SYN 1
DATA 4
TxEN
TxRDY
TxEMT
CE FOR WRITE OF THR
DATA 1
DATA 2
DATA 3
DATA 4
ASYNCHRONOUS MODE
D
A
B
C
DATA 1
DATA 2
DATA 3
DATA 4
TxEN
TxRDY
TxEMT
CE FOR WRITE OF THR
DATA 1
DATA 2
DATA 3
DATA 4
WF02250S

NOTES:
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit.

TxRDY, TxEMT (Shown for 5-bit characters, no parity, 2 stop bits [in asynchronous mode])

Programmable Communications Interface (PCI) SCN2651

TIMING DIAGRAMS (Continued)

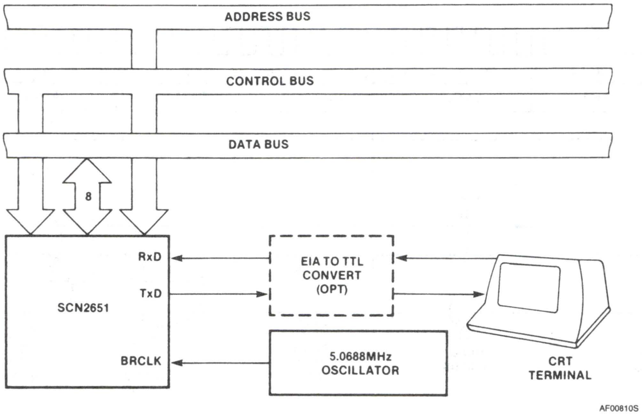


RxC
SYNCHRONOUS MODE
RxD
1 2 3 4 5
SYN 1
DATA 1
DATA 2
DATA 3
DATA 4
DATA 5
IGNORED
RxEN
SYNDET STATUS BIT
RxRDY
CE FOR READ
READ STATUS
READ STATUS
READ RHR (DATA 1)
READ RHR (DATA 2)
READ RHR (DATA 3)
READ RHR (DATA 3)
ASYNCHRONOUS MODE
D
A B C
RxD
DATA 1
DATA 2
DATA 3
DATA 4
RxEN
RxRDY
OVERRUN STATUS BIT
CE FOR READ
READ RHR (DATA 1)
READ RHR (DATA 3)
WF02270S

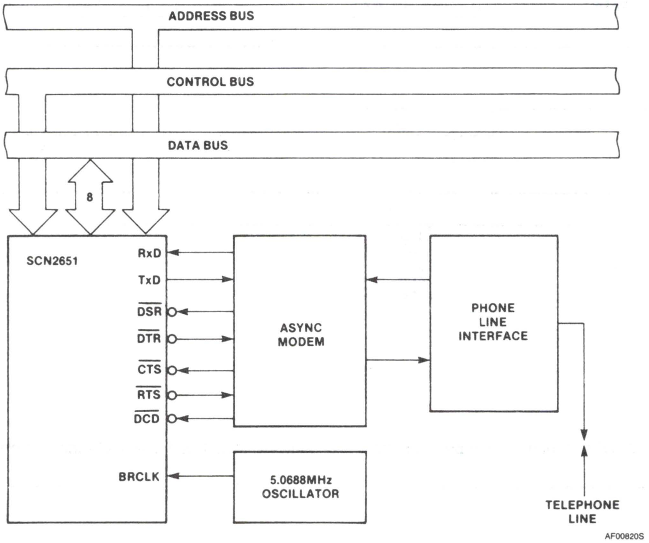
NOTES:
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition

RxRDY (Shown for 5-bit characters, no parity, 2 stop bits [in asynchronous mode])

TYPICAL APPLICATIONS

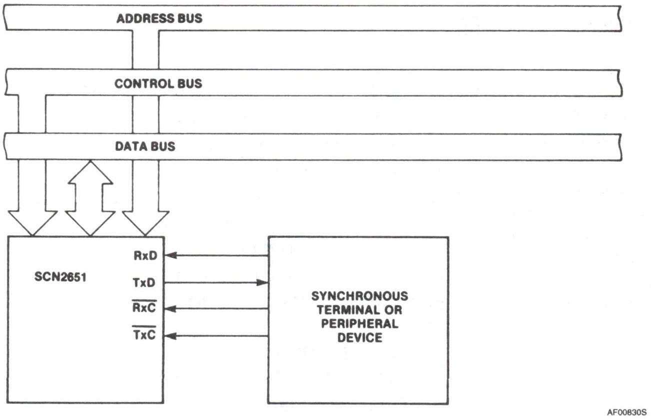

ADDRESS BUS
CONTROL BUS
DATA BUS
8
SCN2651
RxD
TxD
BRCLK
EIA TO TTL CONVERT (OPT)
5.0688MHz OSCILLATOR
CRT TERMINAL
AF00810S

Asynchronous Interface to CRT Terminal

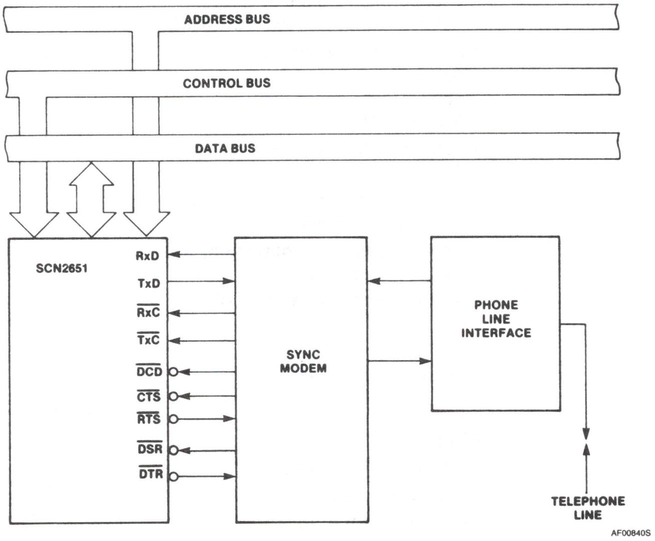

ADDRESS BUS
CONTROL BUS
DATA BUS
8
SCN2651
RxD
TxD
DSR
DTR
CTS
RTS
DCD
BRCLK
ASYNC MODEM
PHONE LINE INTERFACE
5.0688MHz OSCILLATOR
TELEPHONE LINE
AF00820S

Asynchronous Interface to Telephone Lines

TYPICAL APPLICATIONS (Continued)


ADDRESS BUS
CONTROL BUS
DATA BUS
SCN2651
RxD
TxD
RxC
TxC
SYNCHRONOUS TERMINAL OR PERIPHERAL DEVICE
AF00830S

Synchronous Interface to Terminal or Peripheral Device


ADDRESS BUS
CONTROL BUS
DATA BUS
SCN2651
RxD
TxD
RxC
TxC
DCD
CTS
RTS
DSR
DTR
SYNC MODEM
PHONE LINE INTERFACE
TELEPHONE LINE
AF00840S

Synchronous Interface to Telephone Lines

Signetics

SCN2652/SCN68652
Multi-Protocol Communications Controller (MPCC)

Product Specification

Microprocessor Products

DESCRIPTION

The SCN2652/68652 Multi-Protocol Communications Controller (MPCC) is a monolithic n-channel MOS LSI circuit that formats, transmits and receives synchronous serial data while supporting bit-oriented or byte control protocols. The chip is TTL compatible, operates from a single +5V supply, and can interface to a processor with an 8 or 16-bit bidirectional data bus.

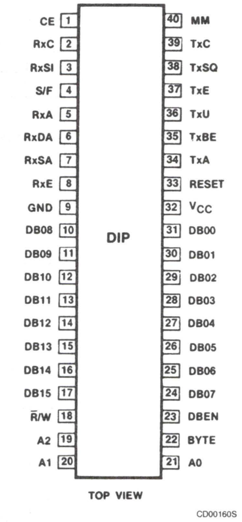
FEATURES

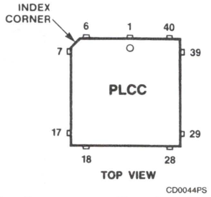
- DC to 1Mbps or 2Mbps data rate
- Bit-oriented protocols (BOP): SDLC, ADCCP, HDLC
- Byte-control protocols (BCP): DDCMP, BISYNC (external CRC)
- Programmable operation
  - 8- or 16-bit 3-State data bus
  - Error control - CRC or VRC or none
  - Character length - 1 to 8 bits for BOP or 5 to 8 bits for BCP
  - SYNC or secondary station address comparison for BCP-BOP
  - Idle transmission of SYNC/FLAG or MARK for BCP-BOP
- Automatic detection and generation of special BOP control sequences, i.e., FLAG, ABORT, GA
- Zero insertion and deletion for BOP
- Short character detection for last BOP data character
- SYNC generation, detection, and stripping for BCP
- Maintenance mode for self-testing
- TTL compatible
- Single +5V supply

APPLICATIONS

- Intelligent terminals
- Line controllers
- Network processors
- Front end communications
- Remote data concentrators
- Communication test equipment
- Computer to computer links

PIN CONFIGURATIONS


CE 1
RxC 2
RxSI 3
S/F 4
RxA 5
RxDA 6
RxSA 7
RxE 8
GND 9
DB08 10
DB09 11
DB10 12
DB11 13
DB12 14
DB13 15
DB14 16
DB15 17
R̄/W 18
A2 19
A1 20
40 MM
39 TxC
38 TxSQ
37 TxE
36 TxU
35 TxBE
34 TxA
33 RESET
32 VCC
31 DB00
30 DB01
29 DB02
28 DB03
27 DB04
26 DB05
25 DB06
24 DB07
23 DBEN
22 BYTE
21 A0
DIP
TOP VIEW
CD00160S


INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

Pin	Function	Pin	Function
1	NC	23	NC
2	CE	24	A0
3	RxC	25	BYTE
4	RxSI	26	DBEN
5	S/F	27	DB07
6	RxA	28	DB06
7	RxDA	29	DB05
8	RxSA	30	DB04
9	RxE	31	DB03
10	GND	32	DB02
12	NC	33	DB01
11	DB08	34	NC
13	DB09	35	DB00
14	DB10	36	V_{CC}
15	DB11	37	RESET
16	DB12	38	TxA
17	DB13	39	TxBE
18	DB14	40	TxU
19	DB15	41	TxE
20	R̄/W	42	TxSQ
21	A2	43	TxC
22	A1	44	MM

NOTE:
0 is least significant bit, highest number (that is, DB15, A2) is most significant bit.

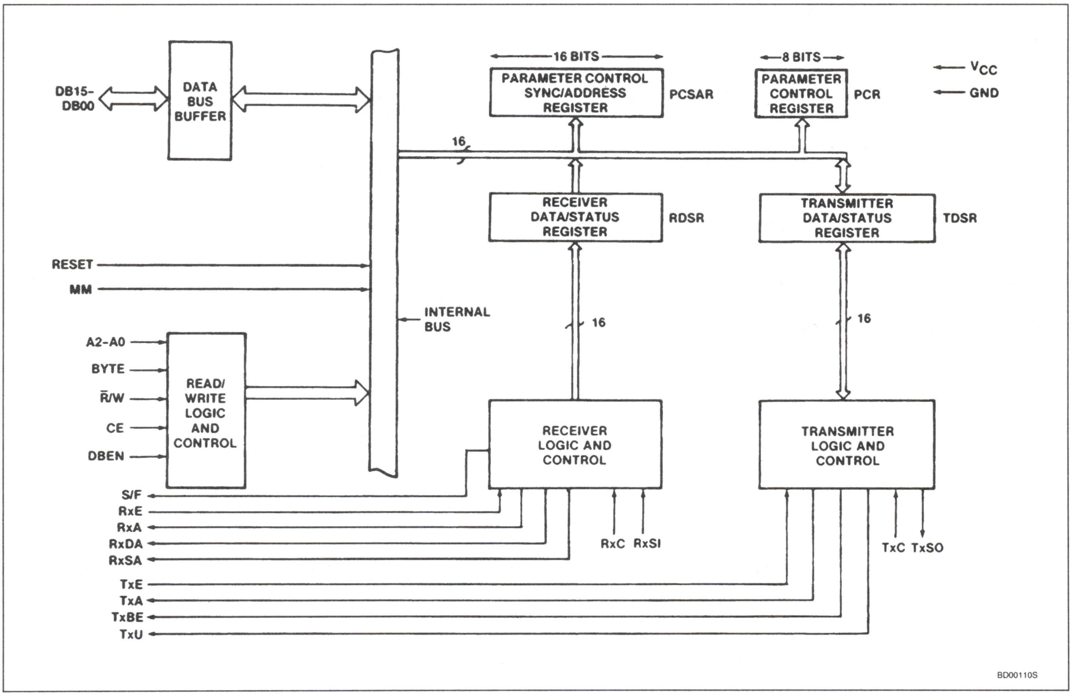
Multi-Protocol Communications Controller (MPCC) SCN2652/SCN68652

ORDERING INFORMATION

PACKAGES		V_{CC} = 5V ±5%		
		Commercial	Automotive	Extended
		0°C to +70°C	-40°C to +85°C	-55°C to +125°C
Ceramic DIP	1MHz	SCN2652AC1I40	Contact Factory	Contact Factory
	2MHz	SCN2652AC2I40	Contact Factory	Contact Factory
Plastic DIP	1MHz	SCN2652AC1N40	Contact Factory	Not Available
	2MHz	SCN2652AC2N40	Contact Factory	Not Available
Plastic LCC	1MHz	SCN2652AC1A44	Contact Factory	Not Available
	2MHz	SCN2652AC2A44	Contact Factory	Not Available

NOTE:
SCN68652 is identical to SCN2652. Order using part numbers shown above.

BLOCK DIAGRAM


DB15-
DB00
DATA BUS BUFFER
16 BITS
PARAMETER CONTROL SYNC/ADDRESS REGISTER
PCSAR
8 BITS
PARAMETER CONTROL REGISTER
PCR
V_{CC}
GND
16
RECEIVER DATA/STATUS REGISTER
RDSR
TRANSMITTER DATA/STATUS REGISTER
TDSR
RESET
MM
INTERNAL BUS
16
16
A2-A0
BYTE
R̅/W
CE
DBEN
READ/ WRITE LOGIC AND CONTROL
RECEIVER LOGIC AND CONTROL
TRANSMITTER LOGIC AND CONTROL
S/F
RxE
RxA
RxDA
RxSA
RxC RxSI
TxC TxSO
TxE
TxA
TxBE
TxU
BD00110S

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
DB15 – DB00	17 – 10 24 – 31	I/O	**Data Bus:** DB07 – DB00 contain bidirectional data while DB15 – DB08 contain control and status information to or from the processor. Corresponding bits of the high and low order bytes can be wire OR'ed onto an 8-bit bus. The data bus is floating if either CE or DBEN are low.
A2 – A0	19 – 21	I	**Address Bus:** A2 – A0 select internal registers. The four 16-bit registers can be addressed on a word or byte basis. See Register Address section.
BYTE	22	I	**Byte:** Single byte (8-bit) data bus transfers are specified when this input is high. A low-level specifies 16-bit data bus transfers.
CE	1	I	**Chip Enable:** A high input permits a data bus operation when DBEN is activated.
$\overline{R}$/W	18	I	**Read/Write:** $\overline{R}$/W controls the direction of data bus transfer. When high, the data is to be loaded into the addressed register. A low input causes the contents of the addressed register to be presented on the data bus.
DBEN	23	I	**Data Bus Enable:** After A2 – A0, CE, BYTE and $\overline{R}$/W are set up, DBEN may be strobed. During a read, the 3-State data bus (DB) is enabled with information for the processor. During a write, the stable data is loaded into the addressed register and TxBE will be reset if TDSR was addressed.
RESET	33	I	**Reset:** A high-level initializes all internal registers (to zero) and timing.
MM	40	I	**Maintenance Mode:** MM internally gates TxSO back to RxSI and TxC to RxC for off line diagnostic purposes. The RxC and RxSI inputs are disabled and TxSO is high when MM is asserted.
RxE	8	I	**Receiver Enable:** A high-level input permits the processing of RxSI data. A low-level disables the receiver logic and initializes all receiver registers and timing.
RxA	5	O	**Receiver Active:** RxA is asserted when the first data character of a message is ready for the processor. In the BOP mode this character is the address. The received address must match the secondary station address if the MPCC is a secondary station. In BCP mode, if strip-SYNC ($PCSAR_{13}$) is set, the first non-SYNC character is the first data character; if strip-SYNC is zero, the character following the second SYNC is the first data character. In the BOP mode, the closing FLAG resets RxA. In the BCP mode, RxA is reset by a low level at RxE.
RxDA*	6	O	**Receiver Data Available:** RxDA is asserted when an assembled character is in $RDSR_L$ and is ready to be presented to the processor. This output is reset when $RDSR_L$ is read.
RxC	2	I	**Receiver Clock:** RxC (1X) provides timing for the receiver logic. The positive going edge shifts serial data into the RxSR from RxSI.
S/F	4	O	**SYNC/FLAG:** S/F is asserted for one RxC clock time when a SYNC or FLAG character is detected.
RxSA*	7	O	**Receiver Status Available:** RxSA is asserted when there is a zero to one transition of any bit in $RDSR_H$ except for RSOM. It is cleared when $RDSR_H$ is read.
RxSI	3	I	**Receiver Serial Input:** RxSI is the received serial data. Mark = '1', space = '0'.
TxE	37	I	**Transmitter Enable:** A high-level input enables the transmitter data path between $TDSR_L$ and TxSO. At the end of a message, a low-level input causes TxSO = 1(mark) and TxA = 0 after the closing FLAG (BOP) or last character (BCP) is output on TxSO.
TxA	34	O	**Transmitter Active:** TxA is asserted after TSOM ($TDSR_8$) is set and TxE is raised. This output will reset when TxE is low and the closing FLAG (BOP) or last character (BCP) has been output on TxSO.
TxBE*	35	O	**Transmitter Buffer Empty:** TxBE is asserted when the TDSR is ready to be loaded with new control information or data. The processor should respond by loading the TDSR which resets TxBE.
TxU*	36	O	**Transmitter Underrun:** TxU is asserted during a transmit sequence when the service of TxBE has been delayed for one character time. This indicates the processor is not keeping up with the transmitter. Line fill depends on $PCSAR_{11}$. TxU is reset by RESET or setting of TSOM ($TDSR_8$), synchronized by the falling edge of TxC.
TxC	39	I	**Transmitter Clock:** TxC (1X) provides timing for the transmitter logic. The positive going edge shifts data out of the TxSR to TxSO.
TxSO	38	O	**Transmitter Serial Output:** TxSO is the transmitted serial data. Mark = '1', space = '0'.
V_{CC}	32	I	**+5V:** Power supply.
GND	9	I	**Ground:** 0V reference ground.

NOTE:
*Indicates possible interrupt signal

Table 1. Glossary

REGISTERS		NO. OF BITS	DESCRIPTION*
Addressable			
PCSAR	Parameter control sync/ address register	16	PCSAR_H and PCR contain parameters common to the receiver and transmitter. PCSAR_L contains a programmable SYNC character (BCP) or secondary station address (BOP).
PCR	Parameter control register	8	
RDSR	Receive data/status register	16	RDSR_H contains receiver status information. RDSR_L = RxDB contains the received assembled character.
TDSR	Transmit data/status register	16	TDSR_H contains transmitter command and status information. TDSRL = TxDB contains the character to be transmitted.
Internal			
CCSR	Control character shift register	8	These registers are used for character assembly (CSSR, HSR, RxSR), disassembly (TxSR), and CRC accumulation/generation (RxCRC, TxCRC).
HSR	Holding shift register	16	
RxSR	Receiver shift register	8	
TxSR	Transmitter shift register	8	
RxCRC	Receiver CRC accumulation register	16	
TxCRC	Transmitter CRC generation register	16	

NOTES:
*H = High byte - bits 15-8
L = Low byte - bits 7-0

FUNCTIONAL DESCRIPTION

The MPCC can be functionally partitioned into receiver logic, transmitter logic, registers that can be read or loaded by the processor, and data bus control circuitry. The register bit formats are shown in Figure 1 while the receiver and transmitter data paths are depicted in Figures 2 and 3.

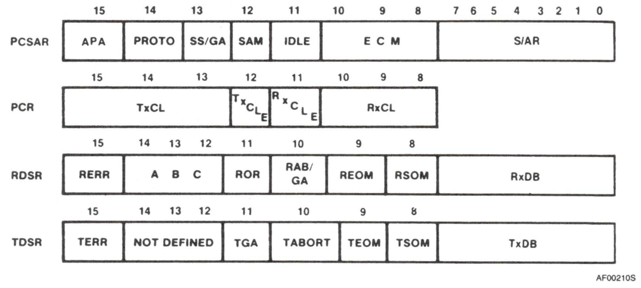
Table 2. Error Control

CHARACTER	DESCRIPTION
FCS	Frame check sequence is transmitted/received as 16 bits following the last data character of a BOP message. The divisor is usually CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) with dividend preset to 1's but can be other wise determined by ECM. The inverted remainder is transmitter as the FCS.
BCC	Block check character is transmitted/received as two successive characters following the last data character of a BCP message. The polynomial is CRC-16 ($X^{16}+X^{15}+X^2+1$) or CRC-CCITT with dividend preset to 0's (as specified by ECM). The true remainder is transmitted as the BCC.

Table 3. Special Characters

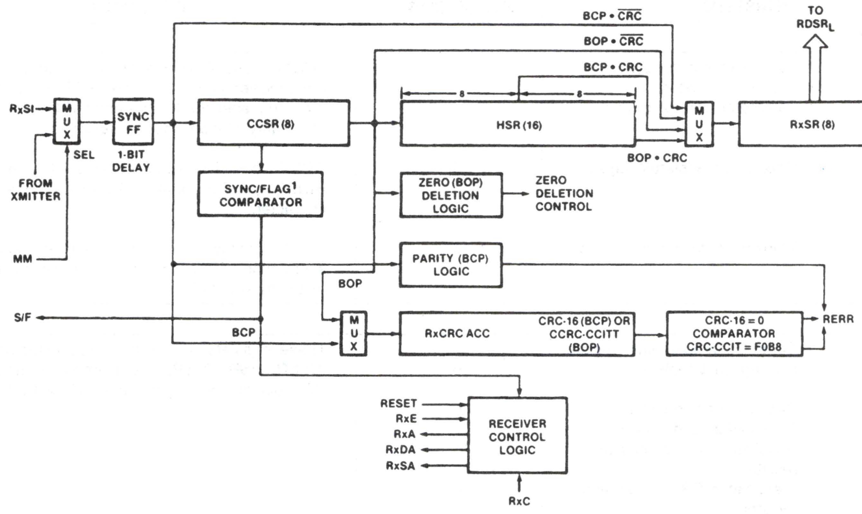
OPERATION	BIT PATTERN	FUNCTION
BOP		
FLAG	01111110	Frame message
ABORT	11111111 generation 01111111 detection	Terminate communication
GA	01111111	Terminate loop mode repeater function
Address	$(\text{PCSAR}_L)^1$	Secondary station address
BCP		
SYNC	(PCSAR_L) or $(\text{TxDB})^2$ generation	Character synchronization

NOTES:
1. ( ) = contents of.
2. For IDLE = 0 or 1 respectively.



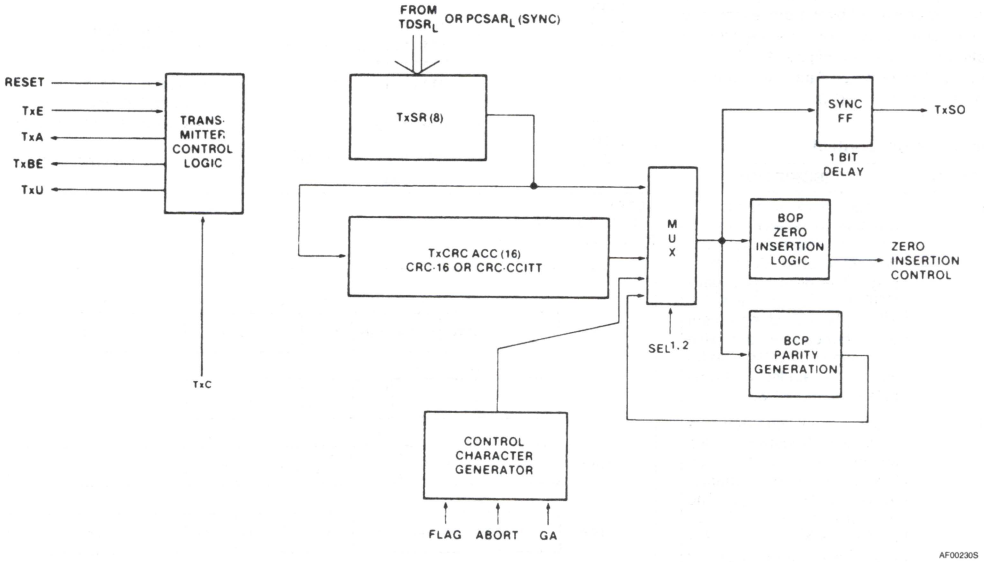
NOTE:
Refer to Register Formats for mnemonics and description.

Figure 1. Short Form Register Bit Formats


BCP • $\overline{CRC}$
BOP • $\overline{CRC}$
BCP • CRC
TO
RDSR$_L$
RxSI
MUX
SYNC FF
1-BIT DELAY
SEL
FROM XMITTER
MM
CCSR (8)
8
8
HSR (16)
MUX
RxSR (8)
BOP • CRC
SYNC/FLAG¹ COMPARATOR
ZERO (BOP) DELETION LOGIC
ZERO DELETION CONTROL
PARITY (BCP) LOGIC
BOP
S/F
BCP
MUX
RxCRC ACC
CRC-16 (BCP) OR CCRC-CCITT (BOP)
CRC-16 = 0 COMPARATOR CRC-CCIT = F0B8
RERR
RESET
RxE
RxA
RxDA
RxSA
RECEIVER CONTROL LOGIC
RxC
AF00220S

NOTES:
1. Detected in SYNC FF and 7 MS bits of CCSR.
2. In BOP mode, a minimum of two data characters must be received to turn the receiver active.

Figure 2. MPCC Receiver Data Path


FROM TDSR$_L$ OR PCSAR$_L$ (SYNC)
RESET
TxE
TxA
TxBE
TxU
TRANS-MITTER CONTROL LOGIC
TxC
TxSR (8)
TxCRC ACC (16) CRC-16 OR CRC-CCITT
MUX
SEL^{1,2}
SYNC FF
1 BIT DELAY
TxSO
BOP ZERO INSERTION LOGIC
ZERO INSERTION CONTROL
BCP PARITY GENERATION
CONTROL CHARACTER GENERATOR
FLAG
ABORT
GA
AF00230S

NOTES:
1. TxCRC selected if TEOM = 1 and the last data character has been shifted out of TxSR.
2. In BCP parity selected will be generated after each character is shifted out of TxSR.

Figure 3. MPCC Transmitter Data Path

RECEIVER OPERATION

General

After initializing the parameter control registers (PCSAR and PCR), the RxE input must be set high to enable the receiver data path. The serial data on the RxSI is synchronized and shifted into an 8-bit Control Character Shift Register (CCSR) on the rising edge of RxC. A comparison between CCSR contents and the FLAG (BOP) or SYNC (BCP) character is made until a match is found. At that time, the S/F output is asserted for one RxC time and the 16-bit Holding Shift Register (HSR) is enabled. The receiver then operates as described below.

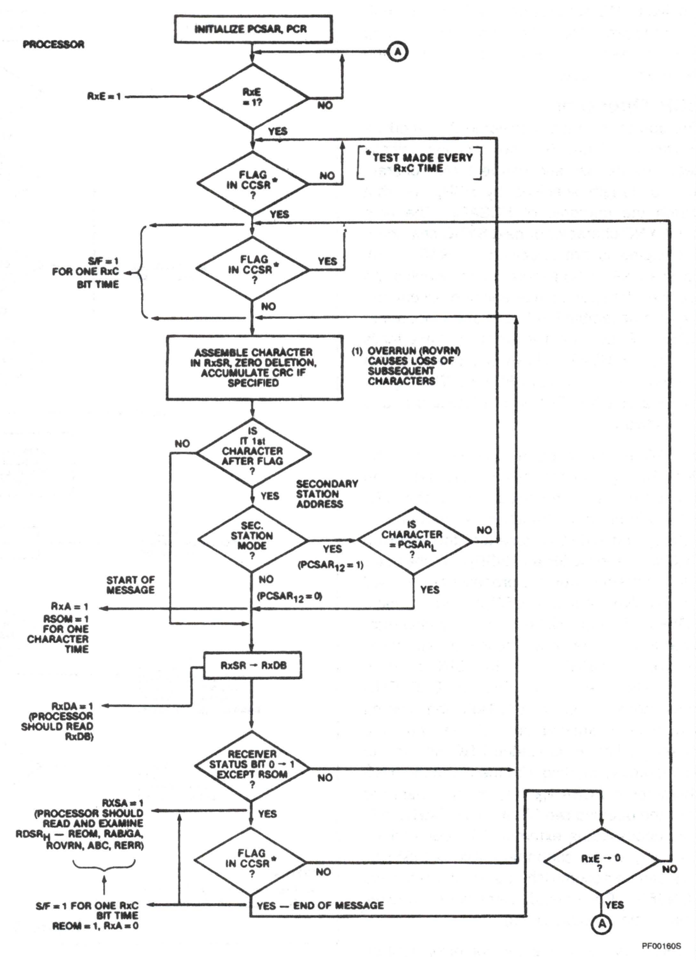

PROCESSOR
INITIALIZE PCSAR, PCR
A
RxE = 1
RxE = 1?
NO
YES
FLAG IN CCSR* ?
NO
*TEST MADE EVERY RxC TIME
YES
S/F = 1 FOR ONE RxC BIT TIME
FLAG IN CCSR* ?
YES
NO
ASSEMBLE CHARACTER IN RxSR, ZERO DELETION, ACCUMULATE CRC IF SPECIFIED
(1) OVERRUN (ROVRN) CAUSES LOSS OF SUBSEQUENT CHARACTERS
IS IT 1st CHARACTER AFTER FLAG ?
NO
YES
SECONDARY STATION ADDRESS
SEC. STATION MODE ?
YES (PCSAR12 = 1)
IS CHARACTER = PCSARL ?
NO
YES
NO (PCSAR12 = 0)
START OF MESSAGE
RxA = 1 RSOM = 1 FOR ONE CHARACTER TIME
RxSR → RxDB
RxDA = 1 (PROCESSOR SHOULD READ RxDB)
RECEIVER STATUS BIT 0 → 1 EXCEPT RSOM ?
NO
YES
RXSA = 1 (PROCESSOR SHOULD READ AND EXAMINE RDSRH — REOM, RAB/GA, ROVRN, ABC, RERR)
FLAG IN CCSR* ?
NO
YES — END OF MESSAGE
S/F = 1 FOR ONE RxC BIT TIME REOM = 1, RxA = 0
RxE → 0 ?
NO
YES
A
PF00160S

Figure 4. BOP Receive

BOP Operation

A flowchart of receiver operation in BOP mode appears in Figure 4. Zero deletion (after five ones are received) is implemented on the received serial data so that a data character will not be interpreted as a FLAG, ABORT, or GA. Bits following the FLAG are shifted through the CCSR, HSR, and into the Receiver Shift Register (RxSR). A character will be assembled in the RxSR and transferred to the $RDSR_L$ for presentation to the processor. At that time the RxDA output will be asserted and the processor must take the character no later than one RxC time after the next character is assembled in the RxSR. If not, an overrun ($RDSR_{11} = 1$) will occur and succeeding characters will be lost.

The first character following the FLAG is the secondary station address. If the MPCC is a secondary station ($PCSAR_{12} = 1$), the contents of RxSR are compared with the address stored in $PCSAR_L$. A match indicates the forthcoming message is intended for the station; the RxA output is asserted, the character is loaded into $RDSR_L$, RxDA is asserted and the Receive Start of Message bit (RSOM) is set. No match indicates that another station is being addressed and the receiver searches for the next FLAG.

If the MPCC is a primary station, ($PCSAR_{12} = 0$), no secondary address check is made; RxA is asserted and RSOM is set once the first non-FLAG character has been loaded into $RDSR_L$ and RxDA has been asserted. Extended address field can be supported by software if $PCSAR_{12} = 0$.

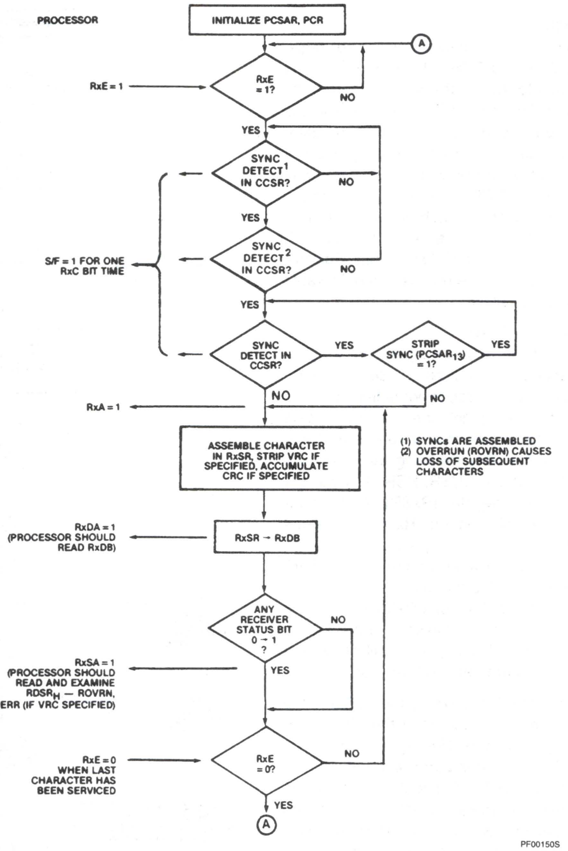
When the 8 bits following the address character have been loaded into $RDSR_L$ and RxDA has been asserted, RSOM will be cleared. The processor should read this 8-bit character and interpret it as the Control field.

Received serial data that follows is read and interpreted as the information field by the processor. It will be assembled into character lengths as specified by PCR_{8-10}. As before, RxDA is asserted each time a character has been transferred into $RDSR_L$ and is cleared when $RDSR_L$ is read by the processor. $RDSR_H$ should only be read when RxSA is asserted. This occurs on a zero to one transition of any bit in $RDSR_H$ except for RSOM. RxSA and all bits in $RDSR_H$ except RSOM are cleared when $RDSR_H$ is read. The processor should check $RDSR_{9-15}$ each time RxSA is asserted. If $RDSR_9$ is set, then $RDSR_{12-15}$ should be examined.

Receiver character length may be changed dynamically in response to RxDA: read the character in RxDB and write the new character length into RxCL. The character length will be changed on the next receiver character boundary. A received residual (short) character will be transferred into RxDB after the previous character in RxDB has been read, i.e. there will not be an overrun. In general the last two characters are protected from overrun.

The CRC-CCITT, if specified by $PCSAR_{8-10}$, is accumulated in RxCRC on each character following the FLAG. When the closing FLAG is detected in the CCSR, the received CRC is in the 16-bit HSR. At that time, the Receive End of Message bit (REOM) will be set; RxSA and RxDA will be asserted. The processor should read the last data character in $RDSR_L$ and the receiver status in $RDSR_{9-15}$. If $RDSR_{15} = 1$, there has been a transmission error; the accumulated CRC-CCITT is incorrect. If $RDSR_{12-14} \neq 0$, the last data charac-

Multi-Protocol Communications Controller (MPCC) SCN2652/SCN68652


PROCESSOR
INITIALIZE PCSAR, PCR
A
RxE = 1
RxE = 1?
NO
YES
SYNC DETECT¹ IN CCSR?
NO
YES
S/F = 1 FOR ONE RxC BIT TIME
SYNC DETECT² IN CCSR?
NO
YES
SYNC DETECT IN CCSR?
YES
STRIP SYNC (PCSAR₁₃) = 1?
YES
NO
NO
RxA = 1
ASSEMBLE CHARACTER IN RxSR, STRIP VRC IF SPECIFIED, ACCUMULATE CRC IF SPECIFIED
(1) SYNCs ARE ASSEMBLED
(2) OVERRUN (ROVRN) CAUSES LOSS OF SUBSEQUENT CHARACTERS
RxDA = 1 (PROCESSOR SHOULD READ RxDB)
RxSR → RxDB
ANY RECEIVER STATUS BIT 0 → 1 ?
NO
YES
RxSA = 1 (PROCESSOR SHOULD READ AND EXAMINE RDSRH — ROVRN, RERR (IF VRC SPECIFIED)
RxE = 0 WHEN LAST CHARACTER HAS BEEN SERVICED
RxE = 0?
NO
YES
A
PF00150S

NOTES:
1. Test made every RxC time.
2. Test made on Rx character boundary.

Figure 5. BCP Receive

ter is not of prescribed length. Neither the received CRC nor closing FLAG are presented to the processor. The processor may drop RxE or leave it active at the end of the received message.

BCP Operation

The operation of the receiver in BCP mode is shown in Figure 5. The receiver initially searches for two successive SYNC characters, of length specified by PCR_{8-10}, that match the contents of $PCSAR_L$. The next non-SYNC character or next SYNC character, if stripping is not specified ($PCSAR_{13} = 0$), causes RxA to be asserted and enables the receiver data path. Once enabled, all characters are assembled in RxSR and loaded into $RDSR_L$. RxDA is active when a character is available in $RDSR_L$. RxSA is active on a 0 to 1 transition of any bit in $RDSR_H$. The signals are cleared when $RDSR_L$ or $RDSR_H$ are read respectively.

If CRC-16 error control is specified by $PCSAR_{8-10}$, the processor must determine the last character received prior to the CRC field. When that character is loaded into $RDSR_L$ and RxDA is asserted, the received CRC will be in CCSR and HSR_L. To check for a transmission error, the processor must read the receiver status ($RDSR_H$) and examine $RDSR_{15}$. This bit will be set for one character time if an error free message has been received. If $RDSR_{15} = 0$, the CRC-16 is in error. The state of $RDSR_{15}$ in BCP CRC mode does not set RxSA. Note that this bit should be examined only at the end of a message. The accumulated CRC will include all characters starting with the first non-SYNC character if $PCSAR_{13} = 1$, or the character after the opening two SYNCs if $PCSAR_{13} = 0$. This necessitates external CRC generation/checking when supporting IBM's BISYNC. This can be accomplished using the Signetics SCN2653 Polynomial Generator/Checker (see Typical Applications).

If VRC has been selected for error control, parity (odd or even) is regenerated on each character and checked when the parity bit is received. A discrepancy causes $RDSR_{15}$ to be set and RxSA to be asserted. This must be sensed by the processor. The received parity bit is stripped before the character is presented to the processor.

When the processor has read the last character of the message, it should drop RxE which disables the receiver logic and initializes all receiver registers and timing.

TRANSMITTER OPERATION

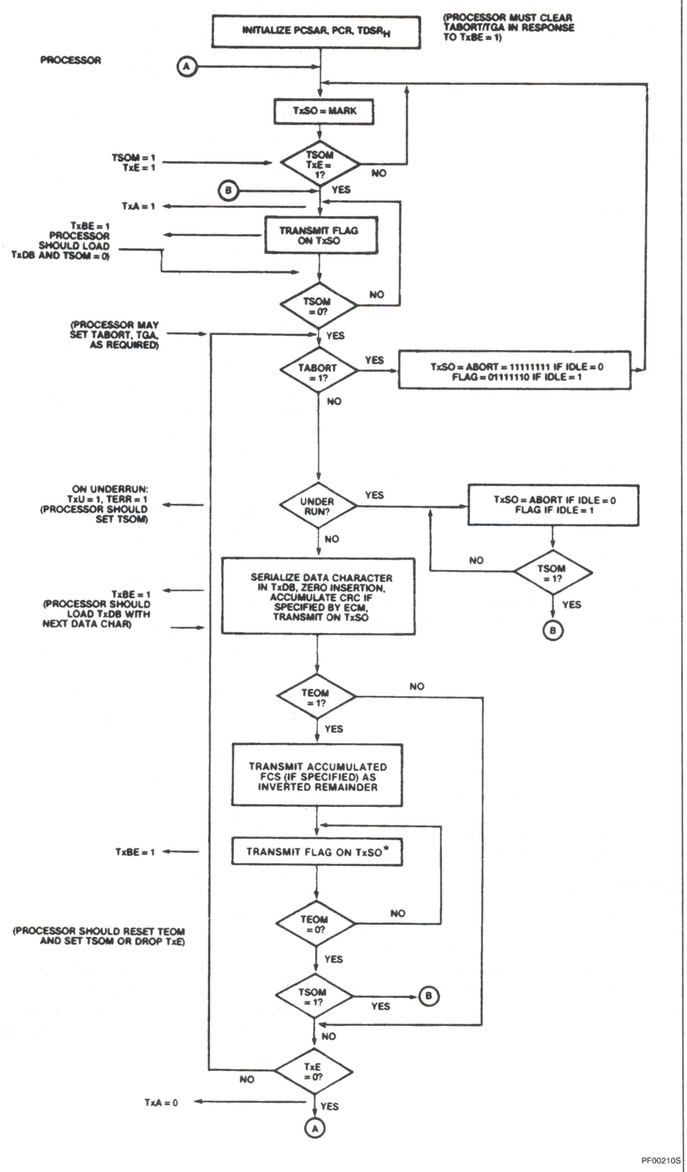
General

After the parameter control registers (PCSAR and PCR) have been initialized, TxSO is held at mark until TSOM ($TDSR_8$) is set and TxE is raised. Then, transmitter operation depends on protocol mode.

BOP Operation

Transmitter operation for BOP is shown in Figure 6. A FLAG is sent after the processor sets the Transmit Start of Message bit (TSOM) and raises TxE. The FLAG is used to synchronize the message that follows. TxA will also be asserted. When TxBE is asserted by the MPCC, the processor should load $TDSR_L$ with the first character of the message. TSOM should be cleared at the same time $TDSR_L$ is loaded (16-bit data bus) or immediately thereafter (8-bit data bus). FLAGs are sent as long as TSOM = 1. For counting the number of FLAGs, the processor should reassert TSOM in response to the assertion of TxBE.

All succeeding characters are loaded into $TDSR_L$ by the processor when TxBE = 1. Each character is serialized in TxSR and transmitted on TxSO. Internal zero insertion logic stuffs a "0" into the serial bit stream after five successive "1s" are sent. This insures a data character will not match a FLAG, ABORT, or GA reserved control character. As each character is transmitted, the Frame Check Sequence (FCS) is generated as specified by Error Control Mode ($PCSAR_{8-10}$). The FCS should be the CRC-CCITT polynomial ($X^{16} + X^{12} + X^5 + 1$) preset to 1s. If an underrun occurs (processor is not keeping up with the transmitter), TxU and TERR ($TDSR_{15}$) will be asserted with ABORT or FLAG used as the TxSO line fill depending on the state of IDLE ($PCSAR_{11}$). The proces-


INITIALIZE PCSAR, PCR, TDSR$_H$
(PROCESSOR MUST CLEAR TABORT/TGA IN RESPONSE TO TxBE = 1)
PROCESSOR
A
TxSO = MARK
TSOM = 1
TxE = 1
TSOM TxE = 1?
NO
YES
B
TxA = 1
TxBE = 1 PROCESSOR SHOULD LOAD TxDB AND TSOM = 0)
TRANSMIT FLAG ON TxSO
TSOM = 0?
NO
YES
(PROCESSOR MAY SET TABORT, TGA, AS REQUIRED)
TABORT = 1?
YES
TxSO = ABORT = 11111111 IF IDLE = 0
FLAG = 01111110 IF IDLE = 1
NO
ON UNDERRUN: TxU = 1, TERR = 1 (PROCESSOR SHOULD SET TSOM)
UNDER RUN?
YES
TxSO = ABORT IF IDLE = 0
FLAG IF IDLE = 1
NO
TSOM = 1?
NO
YES
B
TxBE = 1 (PROCESSOR SHOULD LOAD TxDB WITH NEXT DATA CHAR)
SERIALIZE DATA CHARACTER IN TxDB, ZERO INSERTION, ACCUMULATE CRC IF SPECIFIED BY ECM, TRANSMIT ON TxSO
TEOM = 1?
NO
YES
TRANSMIT ACCUMULATED FCS (IF SPECIFIED) AS INVERTED REMAINDER
TxBE = 1
TRANSMIT FLAG ON TxSO*
TEOM = 0?
NO
(PROCESSOR SHOULD RESET TEOM AND SET TSOM OR DROP TxE)
YES
TSOM = 1?
YES
B
NO
TxE = 0?
NO
TxA = 0
YES
A
PF00210S

NOTE:
*GA will be transmitted if TGA is set together with TEOM.

Figure 6. BOP Transmit

sor must set TSOM to reset the underrun condition. To retransmit the message, the processor should proceed with the normal start of message sequence.

A residual character of 1 to 7 bits may be transmitted at the end of the information field. In response to TxBE, write the residual character length into TxCL and load TxDB with the residual character. Dynamic alteration of character length should be done in exactly the same sequence. The character length will be changed on the next transmit character boundary.

After the last data character has been loaded into TDSR$_L$ and sent to TxSR (TxBE = 1), the processor should set TEOM (TDSR$_9$). The MPCC will finish transmitting the last character followed by the FCS and the closing FLAG. The processor should clear TEOM and drop TxE when the next TxBE is asserted. This corresponds to the start of closing FLAG transmission. When TxE has been dropped. TxA will be low 1½ bit times after the last bit of the closing FLAG has been transmitted. TxSO will be marked after the closing FLAG has been transmitted.

If TxE and TEOM are high, the transmitter continues to send FLAGs. The processor may initiate the next message by resetting TEOM and setting TSOM, or by loading TDSR$_L$ with a data character and then simply resetting TSOM (without setting TSOM).

BCP Operation

Transmitter operation for BCP mode is shown in Figure 7. TxA will be asserted after TSOM = 1 and TxE is raised. At that time SYNC characters are sent from PCSAR$_L$ or TDSR$_L$ (IDLE = 0 or 1) as long as TSOM = 1. TxBE is asserted at the start of transmission of the first SYNC character. For counting the number of SYNCs, the processor should reassert TSOM in response to the assertion of TxBE. When TSOM = 0 transmission is from TDSR$_L$, which must be loaded with characters from the processor each time TxBE is asserted. If this loading is delayed for more than one character time, an underrun results: TxU and TERR are asserted and the TxSO line fill depend on IDLE (PCSAR$_{11}$). The processor must set TSOM and retransmit the message to recover. This is not compatible with IBM's BISYNC, so that the user must not underrun when supporting that protocol.

CRC-16, if specified by PCSAR$_{8-10}$, is generated on each character transmitted from TDSR$_L$ when TSOM = 0. The processor must set TEOM = 1 after the last data character has been sent to TxSR (TxBE = 1). The MPCC will finish transmitting the last data character and the CRC-16 field before sending SYNC characters which are transmitted as long as TEOM = 1. If SYNCs are not desired after CRC-16 transmission, the pro-

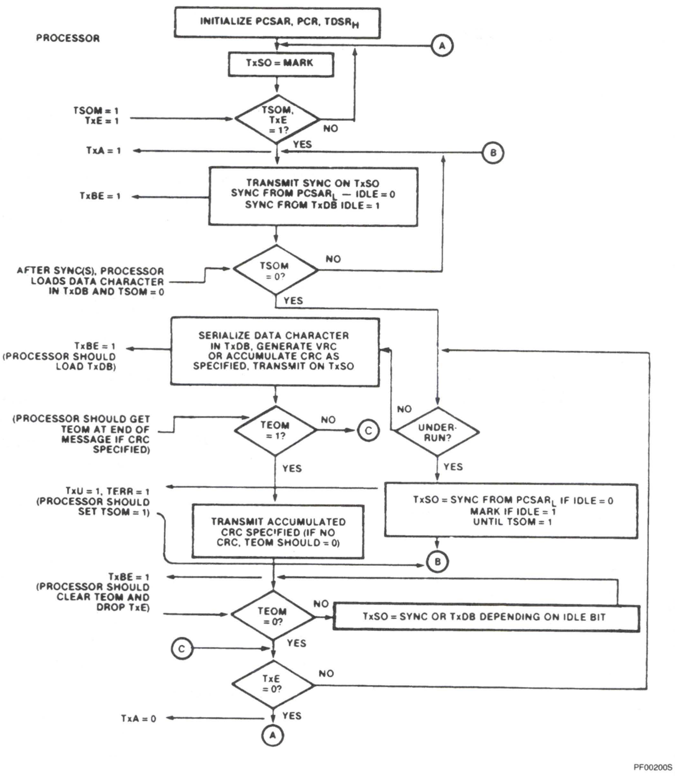

INITIALIZE PCSAR, PCR, TDSR$_H$
PROCESSOR
A
TxSO = MARK
TSOM = 1
TxE = 1
TSOM, TxE = 1?
NO
YES
TxA = 1
B
TxBE = 1
TRANSMIT SYNC ON TxSO
SYNC FROM PCSAR$_L$ — IDLE = 0
SYNC FROM TxDB IDLE = 1
AFTER SYNC(S), PROCESSOR LOADS DATA CHARACTER IN TxDB AND TSOM = 0
TSOM = 0?
NO
YES
TxBE = 1 (PROCESSOR SHOULD LOAD TxDB)
SERIALIZE DATA CHARACTER IN TxDB, GENERATE VRC OR ACCUMULATE CRC AS SPECIFIED, TRANSMIT ON TxSO
(PROCESSOR SHOULD GET TEOM AT END OF MESSAGE IF CRC SPECIFIED)
TEOM = 1?
NO
C
NO
UNDER-RUN?
YES
YES
TxU = 1, TERR = 1 (PROCESSOR SHOULD SET TSOM = 1)
TRANSMIT ACCUMULATED CRC SPECIFIED (IF NO CRC, TEOM SHOULD = 0)
TxSO = SYNC FROM PCSAR$_L$ IF IDLE = 0
MARK IF IDLE = 1
UNTIL TSOM = 1
B
TxBE = 1 (PROCESSOR SHOULD CLEAR TEOM AND DROP TxE)
TEOM = 0?
NO
TxSO = SYNC OR TxDB DEPENDING ON IDLE BIT
YES
C
TxE = 0?
NO
YES
TxA = 0
A
PF00200S

Figure 7. BCP Transmit

cessor should clear TEOM and lower TxE when the TxBE corresponding to the start of CRC-16 transmission is asserted. When TEOM = 0, the line is marked and a new message may be initiated by setting TSOM and raising TxE.

If VRC is specified, it is generated on each data character and the data character length must not exceed 7 bits. For software LRC or CRC, TEOM should be set only if SYNC's are required at the end of the message block.

Special Case

The capability to transmit 16 spaces is provided for line turnaround in half duplex mode or for a control recovery situation. This is achieved by setting TSOM and TEOM, clearing TEOM when TxBE = 1, and proceeding as required.

PROGRAMMING

Prior to initiating data transmission or reception, PCSAR and PCR must be loaded with control information from the processor. The contents of these registers (see Register Format section) will configure the MPCC for the user's specific data communication environment. These registers should be loaded during power-on initialization and after a reset operation. They can be changed at any time that the respective transmitter or receiver is disabled.

The default value for all registers is zero. This corresponds to BOP, primary station mode, 8-bit character length, FCS = CRC-CCITT preset to 1s.

For BOP mode the character length register (PCR) may be set to the desired values during system initialization. The address and control fields will automatically be 8-bits. If a residual character is to be transmitted, TxCL should be changed to the residual character length prior to transmission of that character.

DATA BUS CONTROL

The processor must set up the MPCC register address (A2 – A0), chip enable (CE), byte select (BYTE), and read/write ($\overline{R}$/W) inputs before each data bus transfer operation.

During a read operation ($\overline{R}$/W = 0), the leading edge of DBEN will initiate an MPCC read cycle. The addressed register will place its contents on the data bus. If BYTE = 1, the 8-bit byte is placed on DB15-08 or DB07-00 depending on the H/L status of the register addressed. Unused bits in RDSR$_L$ are zero. If BYTE = 0, all 16 bits (DB15-00) contain MPCC information. The trailing edge of DBEN will reset RxDA and/or RxSA if RDSR$_L$ or RDSR$_H$ is addressed respectively.

DBEN acts as the enable and strobe so that the MPCC will not begin its internal read cycle until DBEN is asserted.

During a write operation ($\overline{R}$/W = 1), data must be stable on DB$_{15-08}$ and/or DB$_{07-00}$ prior to the leading edge of DBEN. The stable data is strobed into the addressed register by DBEN. TxBE will be cleared if the addressed register was TDSR$_H$ or TDSR$_L$.

Table 4. MPCC Register Addressing

	A2	A1	A0	REGISTER
BYTE = 0	**16-BIT DATA BUS = DB_{15} - DB_{00}**			
	0	0	X	RDSR
	0	1	X	TDSR
	1	0	X	PCSAR
	1	1	X	PCR*
BYTE = 1	**8-BIT DATA BUS = $DB_{7\text{-}0}$ or $DB_{15\text{-}8}$\****			
	0	0	0	$RDSR_L$
	0	0	1	$RDSR_H$
	0	1	0	$TDSR_L$
	0	1	1	$TDSR_H$
	1	0	0	$PCSAR_L$
	1	0	1	$PCSAR_H$
	1	1	0	PCR_L*
	1	1	1	PCR_H

NOTES:

* PCR lower byte does not exist. It will be all "0"s when read.

** Corresponding high and low order pins must be tied together.

Table 5. Parameter Control Register (PCR) – (R/W)

BIT	NAME	MODE	FUNCTION
00 – 07	Not Defined		
08 – 10	RxCL	BOP/BCP	Receiver character length is loaded by the processor when RxCLE = 0. The character length is valid after transmission of single byte address and control fields have been received.
11	RxCLE	BOP/BCP	Receiver character length enable should be zero when the processor loads RxCL. The remaining bits of PCR are not affected during loading. Always 0 when read.
12	TxCLE	BOP/BCP	Transmitter character length enable should be zero when the processor loads TxCL. The remaining bits of PCR are not affected during loading. Always 0 when read.
13 – 15	TxCL	BOP/BCP	Transmitter character length is loaded by the processor when TxCLE = 0. Character bit length specification format is identical to RxCL. It is valid after transmission of single byte address and control fields.

RxCL (bits 08 – 10):

10	9	8	Char length (bits)
0	0	0	8
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Table 6. Parameter Control Sync/Address Register (PCSAR)-(R/W)

<table>
<tr><th>BIT</th><th>NAME</th><th>MODE</th><th>FUNCTION</th></tr>
<tr><td>00 – 07</td><td>S/AR</td><td>BOP</td><td>SYNC/address register. Contains the secondary station address if the MPCC is a secondary station. The contents of this register is compared with the first received non-FLAG character to dertermine if the message is meant for this station.</td></tr>
<tr><td></td><td></td><td>BCP</td><td>SYNC character is loaded into this register by the processor. It is used for receive and transmit bit synchronization with bit length specified by RxCL and TxCL.</td></tr>
<tr><td>08 – 10</td><td>ECM</td><td>BOP/BCP</td><td><table>
<tr><th>Error Control Mode</th><th>10</th><th>9</th><th>8</th><th>Suggested Mode</th><th>Char. length</th></tr>
<tr><td>CRC-CCITT preset to 1's</td><td>0</td><td>0</td><td>0</td><td>BOP</td><td>1 – 8</td></tr>
<tr><td>CRC-CCITT preset to 0's</td><td>0</td><td>0</td><td>1</td><td>BCP</td><td>8</td></tr>
<tr><td>Not used</td><td>0</td><td>1</td><td>0</td><td>—</td><td></td></tr>
<tr><td>CRC-16 preset to 0's</td><td>0</td><td>1</td><td>1</td><td>BCP</td><td>8</td></tr>
<tr><td>VRC odd</td><td>1</td><td>0</td><td>0</td><td>BCP</td><td>5 – 7</td></tr>
<tr><td>VRC even</td><td>1</td><td>0</td><td>1</td><td>BCP</td><td>5 – 7</td></tr>
<tr><td>Not used</td><td>1</td><td>1</td><td>0</td><td>—</td><td></td></tr>
<tr><td>No error control</td><td>1</td><td>1</td><td>1</td><td>BCP/BOP</td><td>5 – 8</td></tr>
</table>
ECM should be loaded by the processor during initialization or when both data paths are idle.</td></tr>
<tr><td>11</td><td>IDLE</td><td></td><td>Determines line fill character to be used if transmitter underrun occurs (TxU asserted and TERR set) and transmission of special characters for BOP/BCP.</td></tr>
<tr><td></td><td></td><td>BOP</td><td>IDLE = 0, transmit ABORT characters during underrun and when TABORT = 1.
IDLE = 1, transmit FLAG characters during underrun and when TABORT = 1.</td></tr>
<tr><td></td><td></td><td>BCP</td><td>IDLE = 0 transmit initial SYNC characters and underrun line fill characters from the S/AR.
IDLE = 1 transmit initial SYNC characters from TxDB and marks TxSO during underrun.</td></tr>
<tr><td>12</td><td>SAM</td><td>BOP</td><td>Secondary Address Mode = 1 if the MPCC is a secondary station. This facilitates automatic recognition of the received secondary station address. When transmitting, the processor must load the secondary address into TxDB.
SAM = 0 inhibits the received secondary address comparison which serves to activate the receiver after the first non-FLAG character has been received.</td></tr>
<tr><td>13</td><td>SS/GA</td><td></td><td>Strip SYNC/Go Ahead. Operation depends on mode.</td></tr>
<tr><td></td><td></td><td>BOP</td><td>SS/GA = 1 is used for loop mode only and enables GA detection. When a GA is detected as a closing character, REOM and RAB/GA will be set and the processor should terminate the repeater function. SS/GA = 0 is the normal mode which enables ABORT detection. It causes the receiver to terminate the frame upon detection of an ABORT or FLAG.</td></tr>
<tr><td></td><td></td><td>BCP</td><td>SS/GA = 1, causes the receiver to strip SYNC's immediately following the first two SYNC's detected. SYNC's in the middle of a message will not be stripped. SS/GA = 0, presents any SYNC's after the initial two SYNC's to the processor.</td></tr>
<tr><td>14</td><td>PROTO</td><td></td><td>Determines MPCC Protocol mode</td></tr>
<tr><td></td><td></td><td>BOP</td><td>PROTO = 0</td></tr>
<tr><td></td><td></td><td>BCP</td><td>PROTO = 1</td></tr>
<tr><td>15</td><td>APA</td><td>BOP</td><td>All parties address. If this bit is set, the receiver data path is enabled by an address field of '11111111' as well as the normal secondary station address.</td></tr>
</table>

Table 7. Transmit Data/Status Register (TDSR) (R/W Except TDSR15)

BIT	NAME	MODE	FUNCTION
00 – 07	TxDB	BOP/BCP	Transmit data buffer. Contains processor loaded characters to be serialized in TxSR and transmitted on TxSO.
08	TSOM		Transmitter start of message. Set by the processor to initate message transmission provided TxE = 1.
		BOP	TSOM = 1 generates FLAGs. When TSOM = 0 transmission is from TxDB and FCS generation (if specified) begins. FCS, as specified by PCSAR$_{8-10}$, should be CRC-CCITT preset to 1's.
		BCP	TSOM = 1 generates SYNCs from PCSAR$_L$ or transmits from TxDB for IDLE = 0 or 1 respectively. When TSOM = 0 transmission is from TxDB and CRC generation (if specified) begins.
09	TEOM		Transmit end of message. Used to terminate a transmitted message.
		BOP	TEOM = 1 causes the FCS and the closing FLAG to be transmitted following the transmission of the data character in TxSR. FLAGs are transmitted until TEOM = 0. ABORT or GA are transmitted if TABORT or TGA are set when TEOM = 1.
		BCP	TEOM = 1 causes CRC-16 to be transmitted (if selected) followed by SYNCs from PCSAR$_L$ or TxDB (IDLE = 0 or 1). Clearing TEOM prior to the end of CRC-16 transmission (when TxBE = 1) causes TxSO to be marked following the CRC – 16. TxE must be dropped before a new message can be initiated. If CRC is not selected, TEOM should not be set.
10	TABORT	BOP	Transmitter abort = 1 will cause ABORT or FLAG to be sent (IDLE = 0 or 1) after the current character is transmitted. (ABORT = 11111111)
11	TGA	BOP	Transmit go ahead (GA) instead of FLAG when TEOM = 1. This facilitates repeater termination in loop mode. (GA = 01111111)
12 – 14	Not Defined		
15	TERR	Read only	Transmitter error = 1 indicates the TxDB has not been loaded in time (one character time-½ TxC period after TxBE is asserted) to maintain continuous transmission. TxU will be asserted to inform the processor of this condition. TERR is cleared by setting TSOM. See timing diagram.
		BOP	ABORT's or FLAG's are sent as fill characters (IDLE = 0 or 1)
		BCP	SYNC's or MARK's are sent as fill characters (IDLE = 0 or 1). For IDLE = 1 the last character before underrun is not valid.

Table 8. Receiver Data/Status Register (RDSR) – (Read Only)

BIT	NAME	MODE	FUNCTION
00 – 07	RxDB	BOP/BCP	Receiver data buffer. Contains assembled characters from the RxSR. If VRC is specified, the parity bit is stripped.
08	RSOM	BOP	Receiver start of message = 1 when a FLAG followed by a non-FLAG has been received and the latter character matches the secondary station if SAM = 1. RxA will be asserted when RSOM = 1. RSOM resets itself after one character time and has no affect on RxSA.
09	REOM	BOP	Receiver end of message = 1 when the closing FLAG is detected and the last data character is loaded into RxDB or when an ABORT/GA character is received. REOM is cleared on reading $RDSR_H$, reset operation, or dropping of RxE.
10	RAB/GA	BOP	Received ABORT or GA character = 1 when the receiver senses an ABORT character if SS/GA = 0 or a GA character if SS/GA = 1. RAB/GA is cleared on reading $RDSR_H$, reset operation, or dropping of RxE. A received abort does not set RxDA.
11	ROR	BOP/BCP	Receiver overrun = 1 indicates the processor has not read last character in the RxDB within one character time + ½ RxC period after RxDA is asserted. Subsequent characters will be lost. ROR is cleared on reading $RDSR_H$, reset operation, or dropping of RxE.
12 – 14	ABC	BOP	Assembled bit count. Specifies the number of bits in the last received data character of a message and should be examined by the processor when REOM = 1(RxDA and RxSA asserted). ABC = 0 indicates the message was terminated (by a flag or GA) on a character boundary as specified by PCR_{8-10}. Otherwise, ABC = number of bits in the last data character. ABC is cleared when $RDSR_H$ is read, reset operation, or dropping RxE. The residual character is right justified in $RDSR_L$.
15	RERR	BOP/BCP	Receiver error indicator should be examined by the processor when REOM = 1 in BOP, or when the processor determines the last data character of the message in BCP with CRC or when RxSA is set in BCP with VRC. CRC-CCITT preset to 1's/0's as specified by $PCSAR_{8-10}$: RERR = 1 indicates FCS error (CRC ≠ F0B8 or ≠ 0) RERR = 0 indicates FCS received correctly (CRC = F0B8 or = 0) CRC-16 preset to 0's on 8-bit characters specified by $PSCAR_{8-10}$: RERR = 1 indicates CRC-16 received correctly (CRC = 0). RERR = 0 indicates CRC-16 error (CRC≠0) VRC specified by $PCSAR_{8-10}$: RERR = 1 indicates VRC error RERR = 0 indicates VRC is correct.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	Note 4	°C
T_{STG}	Storage temperature range Input or output voltages with respect to GND[3]	−65 to +150 −0.3 to +15	°C V
V_{CC}	With respect to GND	−0.3 to +7	V

DC ELECTRICAL CHARACTERISTICS[4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Input voltage						
V_{IL} V_{IH}	Low High		 2		0.8	V
Output voltage						
V_{OL} V_{OH}	Low High	$I_{OL} = 1.6mA$ $I_{OH} = -100\mu A$	 2.4		0.4	V
I_{CC}	Power supply current	$V_{CC} = 5.25V$, $T_A = 0°C$			150	mA
Leakage current						
I_{IL} I_{OL}	Input Output	$V_{IN} = 0$ to 5.25V $V_{OUT} = 0$ to 5.25V			10 10	µA
Capacitance						
C_{IN} C_{OUT}	Input Output	$V_{IN} = 0V$, f = 1MHz $V_{OUT} = 0V$, f = 1MHz			20 20	pF

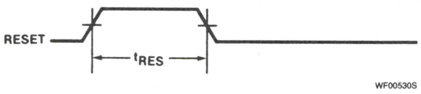
AC ELECTRICAL CHARACTERISTICS[4, 5, 6]

SYMBOL	PARAMETER	1MHz CLOCK VERSION			2MHz CLOCK VERSION			UNIT
		Min	Typ	Max	Min	Typ	Max	
Setup and hold time								
t_{ACS}	Address/control setup	50			50			ns
t_{ACH}	Address/control hold	0			0			
t_{DS}	Data bus setup (write)	50			50			
t_{DH}	Data bus hold (write)	0			0			
t_{RXS}	Receiver serial data setup	150			150			
t_{RxH}	Receiver serial data hold	150			150			
Pulse width								
t_{RES}	RESET	250			250			ns
t_{DBEN}	DBEN	250		m[7]	200		m[7]	
Delay Time								
t_{DD}	Data bus (read)			200			170	ns
t_{TxD}	Transmit serial data			325			250	
t_{DBEND}	DBEN to DBEN delay	200			200			
t_{DF}	Data bus float time (read)			150			150	ns
f	Clock (RxC,TxC) frequency			1			2	MHz
t_{CLK1}	Clock high (MM = 0)	340			165			ns
t_{CLK2}	Clock high (MM = 1)	490			240			
t_{CLKO}	Clock low	490			240			

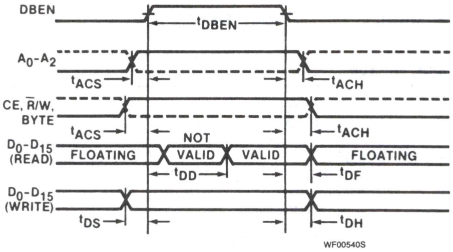
NOTES:

1. Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation sections of this specification is not implied.
2. For operating at elevated temperatures the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified. See ordering code table for applicable temperature range and operating supply range.
5. All voltage measurements are referenced to ground. All time measurements are at 0.8V or 2V. Input voltage levels for testing are 0.4V and 2.4V.
6. Output load $C_L = 100pF$.
7. m = TxC low and applies to writing to $TDSR_H$ only.

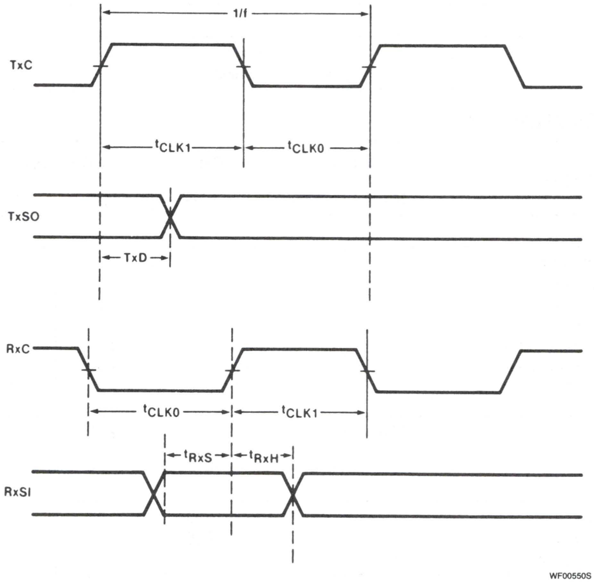
TIMING DIAGRAMS


RESET
t_{RES}
WF00530S

Reset

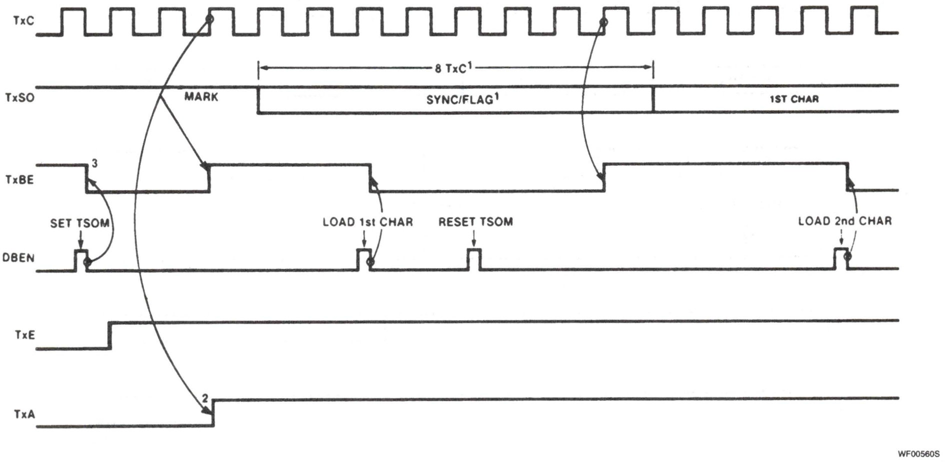

DBEN
t_{DBEN}
A_0-A_2
t_{ACS}
t_{ACH}
CE, $\overline{R}$/W, BYTE
t_{ACS}
t_{ACH}
D_0-D_{15} (READ)
FLOATING
NOT VALID
VALID
FLOATING
t_{DD}
t_{DF}
D_0-D_{15} (WRITE)
t_{DS}
t_{DH}
WF00540S

Read and Write Data Bus


1/f
TxC
t_{CLK1}
t_{CLK0}
TxSO
TxD
RxC
t_{CLK0}
t_{CLK1}
t_{RxS}
t_{RxH}
RxSI
WF00550S

Clock

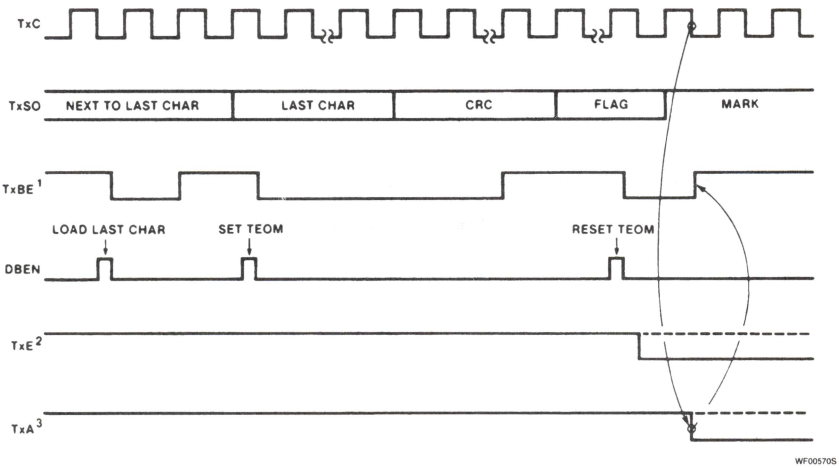
TIMING DIAGRAMS (Continued)


TxC
TxSO
MARK
8 TxC¹
SYNC/FLAG¹
1ST CHAR
TxBE
3
SET TSOM
LOAD 1st CHAR
RESET TSOM
LOAD 2nd CHAR
DBEN
TxE
TxA
2
WF00560S

NOTES:

1. SYNC may be 5 to 8 bits and will contain parity bit as specified.
2. TxA goes high relative to TxC rising edge after TSOM has been set and TxE has been raised.
3. TxBE goes low relative to DBEN falling edge on the first write transfer into TDSR. It is reasserted 1 TxC time before the first bit of the transmitted SYNC/FLAG. TxBE then goes low relative to DBEN falling edge when writing into $TDSR_H$ and/or $TDSR_L$. It is reasserted on the rising edge of the TxC that corresponds to the transmission of the last bit of each character, except in BOP mode when the CRC is to be sent as the next character (see Transmit Timing - End of Message).

Transmit — Start of Message

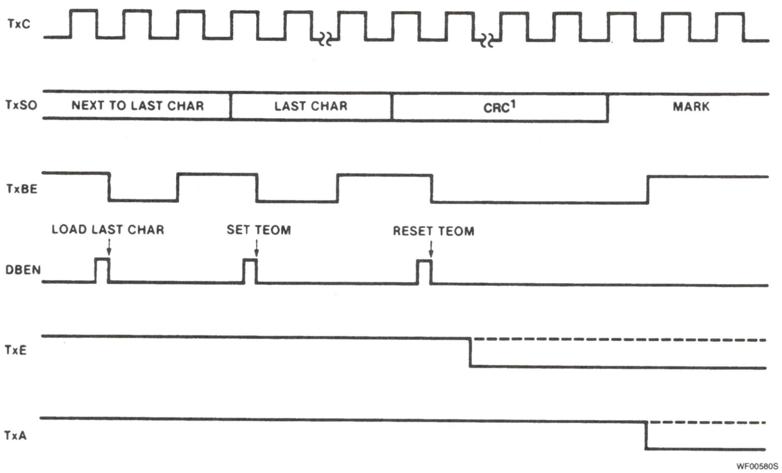

TxC
TxSO
NEXT TO LAST CHAR
LAST CHAR
CRC
FLAG
MARK
TxBE¹
LOAD LAST CHAR
SET TEOM
RESET TEOM
DBEN
TxE²
TxA³
WF00570S

NOTES:

1. TxBE goes low relative to the falling edge of DBEN corresponding to loading $TDSR_{H/L}$. It goes high one TxC before character transmission begins and also when TxA has been dropped.
2. TxE can be dropped before resetting TEOM if TxBE (corresponding to the closing FLAG) is high. Alternatively TxE can remain high and a new message initiated.
3. TxA goes low after TxE has been dropped and 1½ TxC's after the last bit of the closing FLAG has been transmitted.

Transmit — End of BOP Message

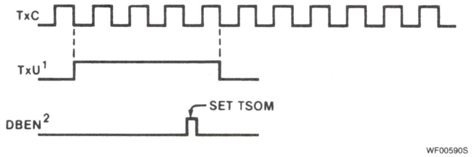
TIMING DIAGRAMS (Continued)


TxC
TxSO
NEXT TO LAST CHAR
LAST CHAR
CRC1
MARK
TxBE
LOAD LAST CHAR
SET TEOM
RESET TEOM
DBEN
TxE
TxA
WF00580S

NOTE:

1. When SCN2652 generated CRC is not required, TEOM should only be set if SYNCs are to follow the message block. In that case, TxE should be dropped in response to TxBE (which corresponds to the start of transmission of the last character). When CRC is required, TxE must be dropped before CRC transmission is complete. Otherwise, the contents of TxDB will be shifted out on TxSO. This facilitates transmission of contiguous messages.

Transmit Timing — End of BCP Message

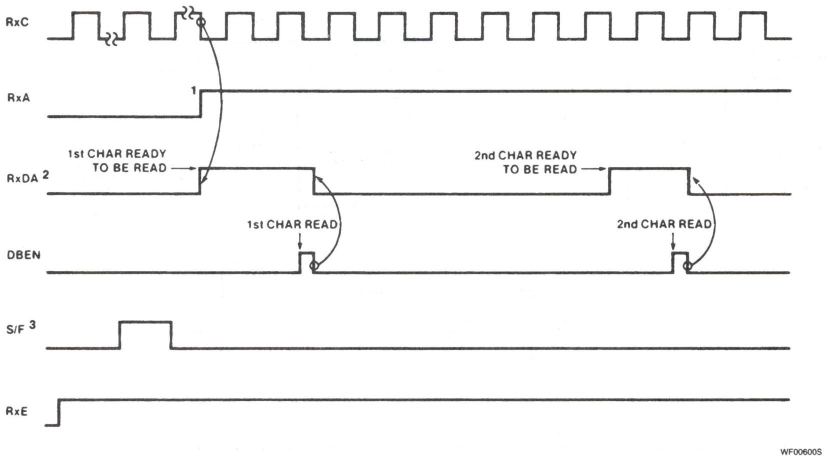

TxC
TxU1
SET TSOM
DBEN2
WF00590S

NOTES:

1. TxU goes active relative to TxC falling edge if TxBE has not been serviced after n-½ TxC times (where n = transmit character length). TxU is reset on the TxC falling edge following assertion of the TSOM command.
2. An underrun will occur at the next character boundary if TEOM is reset and the transmitter remains enabled, unless the TSOM command is asserted or a character is loaded into the TxDB.

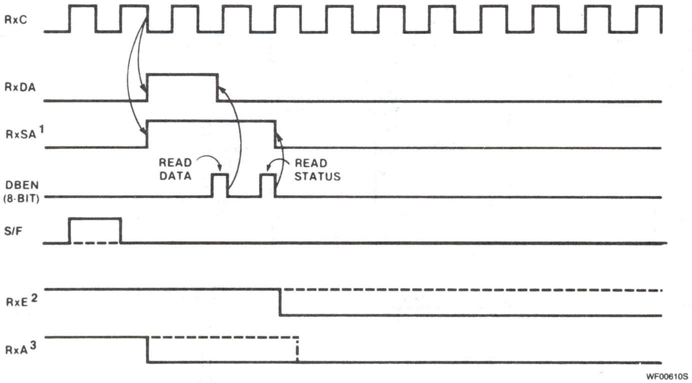
Transmit Underrun

TIMING DIAGRAMS (Continued)


RxC
RxA
1
1st CHAR READY TO BE READ
RxDA 2
2nd CHAR READY TO BE READ
1st CHAR READ
2nd CHAR READ
DBEN
S/F 3
RxE
WF00600S

NOTES:
1. RxA goes high relative to falling edge of RxC when RxE is high and: a. A data character following two SYNC's is in RxDB (BCP mode). b. Character following FLAG is in RxDB (BOP primary station mode). c. Character following FLAG is in RxDB and character matches the secondary station address or all parties address (BOP secondary station mode).
2. RxDA goes high on RxC falling edge when a character in RxDB is ready to be read. It comes up before RxSA and goes low on the falling edge of DBEN when RxDB is read.
3. S/F goes high relative to rising edge of RxC anytime a SYNC (BCP) or FLAG (BOP) is detected.

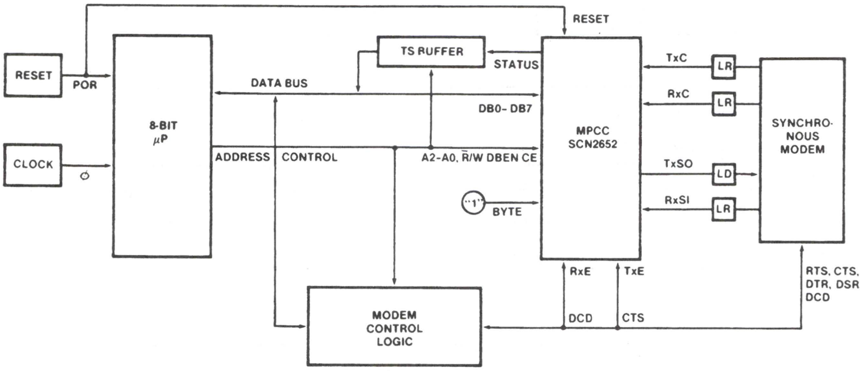
Receive — Start of Message


RxC
RxDA
RxSA 1
READ DATA
READ STATUS
DBEN (8-BIT)
S/F
RxE 2
RxA 3
WF00610S

NOTES:
1. At the end of a BOP message, RxSA goes high when FLAG detection (S/F 1) forces REOM to be set. Processor should read the last data character ($RDSR_L$) and status ($RDSR_H$) which resets RxDA and RxSA respectively. For BCP end of message, RxSA may not be set and S/F = 0. The processor should read the last data character and status.
2. RxE must be dropped for BCP with non-contiguous messages. It may be left on at the end of a BOP message (see BOP Receive Operation).
3. RxA is reset relative to the falling edge of RxC after the closing FLAG of a BOP message (REOM = 1 and RxSA active.) or when RxE is dropped.

Receive — End of Message

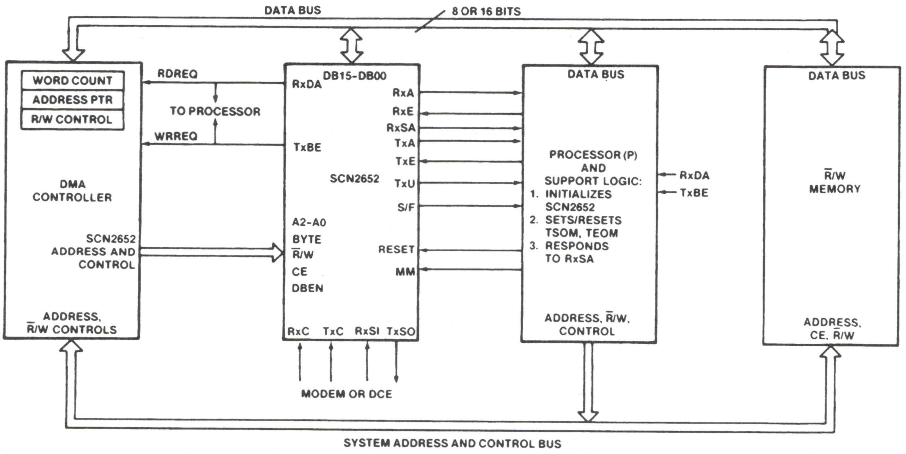
TYPICAL APPLICATIONS


RESET
RESET
POR
CLOCK
Φ
8-BIT
μP
DATA BUS
TS BUFFER
STATUS
DB0- DB7
ADDRESS
CONTROL
A2-A0, R̅/W DBEN CE
"1"
BYTE
MPCC
SCN2652
TxC
RxC
TxSO
RxSI
LR
LR
LD
LR
SYNCHRO-
NOUS
MODEM
RxE
TxE
RTS, CTS,
DTR, DSR,
DCD
MODEM
CONTROL
LOGIC
DCD
CTS
AF00240S

NOTES:

1. Possible μP interrupt requests are: RxDA RxSA TxBE TxU
2. Other SCN2652 status signals and possible uses are: S F line idle indicator, frame delimiter. RxA handshake on RxE, line turn around control. TxA handshake on TxE, line turn around control.
3. Line drivers/receivers (LD/LR) convert EIA to TTL voltages and vice-versa.
4. RTS should be dropped after the CRC (BCP) or FLAG (BOP) has been transmitted. This forces CTS low and TxE low.
5. Corresponding high and low order bits of DB must be OR tied.

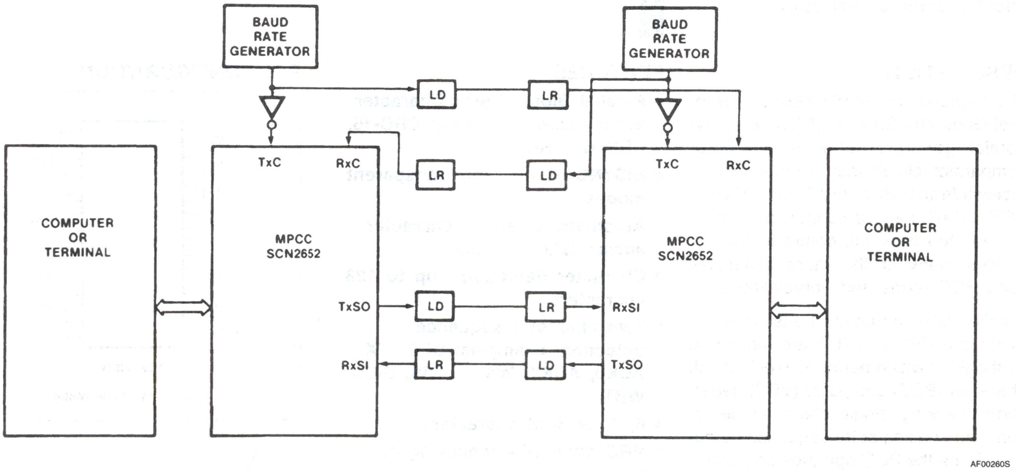
SCN2652 MPCC Microprocessor Interface


DATA BUS
8 OR 16 BITS
WORD COUNT
ADDRESS PTR
R/W CONTROL
DMA
CONTROLLER
SCN2652
ADDRESS AND
CONTROL
ADDRESS,
R̅/W CONTROLS
RDREQ
TO PROCESSOR
WRREQ
RxDA
DB15-DB00
RxA
RxE
RxSA
TxBE
TxA
TxE
SCN2652
TxU
S/F
A2-A0
BYTE
R̅/W
CE
DBEN
RESET
MM
RxC
TxC
RxSI
TxSO
MODEM OR DCE
DATA BUS
PROCESSOR (P)
AND
SUPPORT LOGIC:
1. INITIALIZES
SCN2652
2. SETS/RESETS
TSOM, TEOM
3. RESPONDS
TO RxSA
RxDA
TxBE
ADDRESS, R̅/W,
CONTROL
DATA BUS
R̅/W
MEMORY
ADDRESS,
CE, R̅/W
SYSTEM ADDRESS AND CONTROL BUS
AF00250S

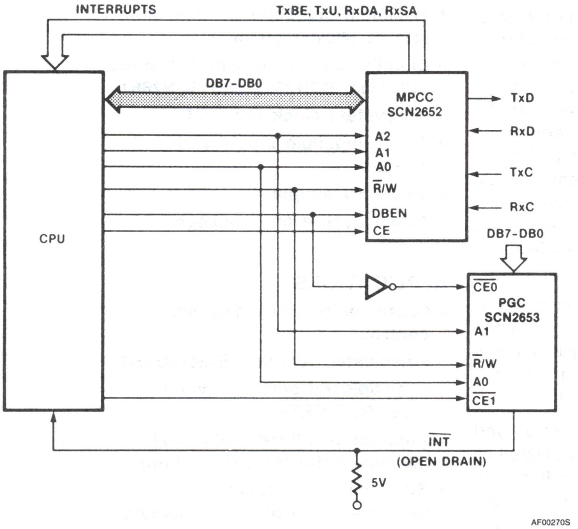
NOTE:
For non-DMA operation TxBE and RxDA are sent to the processor which then loads or reads data characters as required.

DMA/Processor Interface

TYPICAL APPLICATIONS (Continued)


BAUD RATE GENERATOR
BAUD RATE GENERATOR
LD
LR
LR
LD
TxC
RxC
TxC
RxC
COMPUTER OR TERMINAL
MPCC SCN2652
MPCC SCN2652
COMPUTER OR TERMINAL
TxSO
LD
LR
RxSI
RxSI
LR
LD
TxSO
AF00260S

Channel Interface
No Modem — DC Baseband Transmission


INTERRUPTS
TxBE, TxU, RxDA, RxSA
DB7-DB0
MPCC SCN2652
TxD
RxD
TxC
RxC
A2
A1
A0
R̅/W
DBEN
CE
CPU
DB7-DB0
C̅E̅0
PGC SCN2653
A1
R̅/W
A0
C̅E̅1
I̅N̅T̅ (OPEN DRAIN)
5V
AF00270S

SCN2652/SCN2653 Interface Typical Protocols:
BISYNC, DDCMP, SDLC, HDLC

Signetics

SCN2653/SCN68653
Polynomial Generator Checker (PGC)

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics SCN2653/68653 Polynomial Generator Checker (PGC) is a polynomial generator checker/character comparator circuit that complements a receiver/transmitter (R/T or USART/USRT/UART) in the support of character oriented data link controls. Table 1 defines many of the more commonly used PGC terms and abbreviations.

Parallel data characters transferred between the CPU and R/T are monitored by the PGC which performs block check character (BCC) and parity (VRC) generation/checking, single character detection, and two character sequence detection. Since the PGC operates on parallel characters, the data transmission format may be serial (synchronous or asynchronous) or parallel.

There are four modes of BCC accumulation and each mode can select one of three polynomials to compute the BCC. In the BISYNC normal and transparent modes, the PGC determines which characters are to be accumulated and which characters are to be excluded from the accumulation. The block terminating characters and the initiation and termination of BISYNC transparent text can be detected and an interrupt generated. The single interrupt output represents the inclusive OR of four maskable status conditions.

In the automatic accumulation mode, all characters are accumulated while the single accumulate mode requires a specific accumulation command for each character to be accumulated.

Character accumulation control and character comparisons are facilitated by a character class array which places each of 128 characters into one of four character classes. The four classes are normal, SYN/BISYNC not included, block terminating character (BTC)/search character (SC), and secondary search character (SSC).

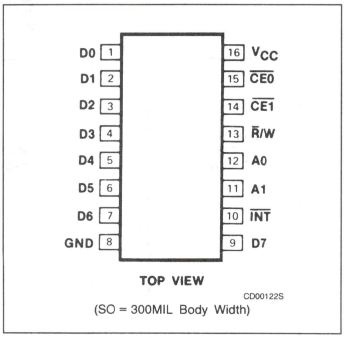
FEATURES

- **Parallel Block Check Character accumulation/checking: CRC-16, CRC-12, LRC-8**
- **BISYNC normal and transparent modes**
- **Automatic or single character accumulation modes**
- **Character detection - up to 128 characters**
- **Two character sequence detection; examples: DLE-STX, ACK0, ACK1, WACK, RVI, DISC, WBT**
- **6, 7, or 8-bit characters**
- **VRC generation/checking on data bus**
- **Four maskable interrupt conditions**
- **Four classes of characters**
- **Internal power-on reset**
- **Maximum character accumulation rate of 500kHz (4Mbps)**
- **Directly compatible with Signetics SCN2651, SCN2652 and SCN2661**
- **No system clock required**
- **TTL compatible inputs and outputs**
- **Single 5V supply**
- **16-pin dual in line package**

APPLICATIONS

- **Character oriented data link control:**
  - **dedicated to one USART/USRT**
  - **multiplexed among several USART/USRTs**
- **Automated BISYNC with 2661 (minimal software intervention)**
- **BCC and VRC generation/detection on a block of memory or peripheral data**
- **Programmable character array comparator**

PIN CONFIGURATION



(SO = 300MIL Body Width)

Polynomial Generator Checker (PGC)

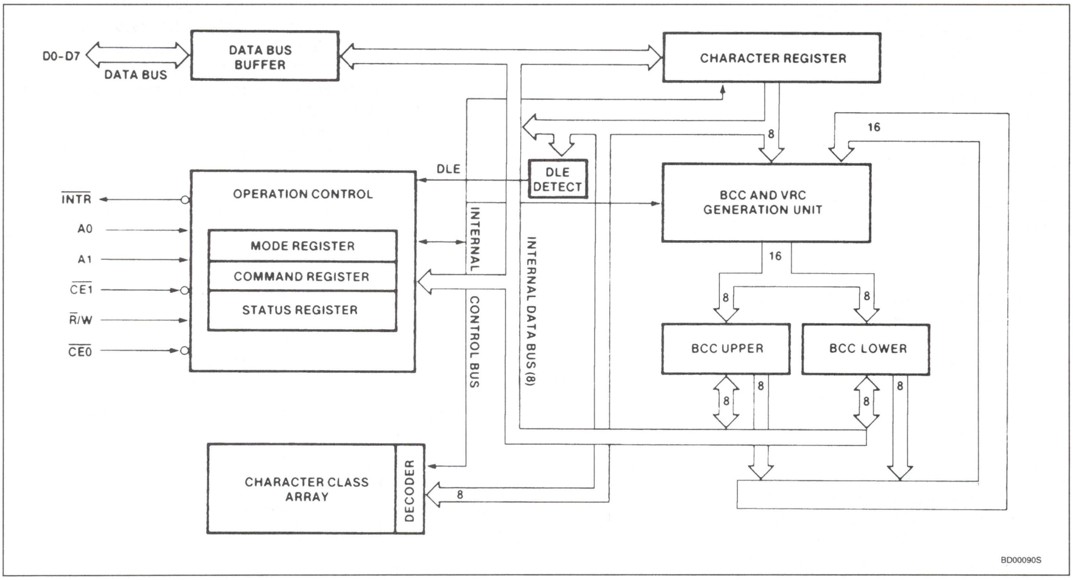
SCN2653/SCN68653

ORDERING INFORMATION

PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to +70°C
Ceramic DIP	SCN2653AC4I16
Plastic DIP	SCN2653AC4N16
Small Outline (SO)	SCN2653ACD16

NOTE:
SCN68653 is identical to SCN2653. Order using numbers shown above.

BLOCK DIAGRAM


D0-D7
DATA BUS
DATA BUS BUFFER
CHARACTER REGISTER
16
8
DLE
DLE DETECT
BCC AND VRC GENERATION UNIT
INTR
A0
A1
CE1
R/W
CE0
OPERATION CONTROL
MODE REGISTER
COMMAND REGISTER
STATUS REGISTER
INTERNAL CONTROL BUS
INTERNAL DATA BUS (8)
16
8
8
BCC UPPER
BCC LOWER
8
8
8
8
CHARACTER CLASS ARRAY
DECODER
8
BD00090S

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V_{CC}	16	I	**+5V:** Power supply
GND	8	I	Ground
A1-A0	11,12	I	**Address Lines:** Used to select internal PGC registers or character class array.
R̅/W	13	I	**Read/Write:** Read command when low, write command when high.
C̅E̅0̅	15	I	**Chip Enable:** Connected to chip enable input of a receiver/transmitter (R/T) circuit. It is used to strobe data being transferred between the CPU and the R̅/T into the PGC character register.
C̅E̅1̅	14	I	**Chip Enable:** Used in conjunction with the R̅/W signal to enable the transfer of data between the PGC and the CPU or DMA controller and to initialize the PGC registers.
D7-D0	9,7-1	I/O	**Data Bus:** 8-bit 3-State bidirectional bus used to transfer data to or from the PGC via C̅E̅0̅ or C̅E̅1̅. All data, mode words, command words, and status information are transferred on this bus. D0 is the least significant bit; D7 is the most significant bit.
I̅N̅T̅	10	O	**Interrupt:** Open drain active low interrupt output that signals the CPU that one or more maskable conditions are true: BCC error, VRC error, BTC/SC detect, SSC detect. The true conditions can be determined by reading the status register which in turn deactivates I̅N̅T̅. A power on, clear BCC, or master reset command causes I̅N̅T̅ to be inactive (high).

Polynomial Generator Checker (PGC) SCN2653/SCN68653

Additional PGC applications include off-line R/T operation where the BCC is generated on data not sent to the R/T, BCC multiplexing by sharing the PGC among several R/Ts and reading/writing the partial BCC accumulation on a character by character basis, VRC generation/checking on characters appearing on a bidirectional data bus, and programmable character comparisons or searches.

PGC operation is half duplex (either receive or transmit, one way or two way alternate). Full duplex (two way simultaneous) is achieved by using two PGCs. The device is directly compatible with the Signetics SCN2651 Programmable Communications interface (PCI) and SCN2661 Enhanced Programmable Communications interface (EPCI). When used in BISYNC modes with the SCN2661, software requirements are minimized by the SCN2653-SCN2661 control character comparisons, character sequence comparisons, and automatic DLE insertion/detection.

Other bus oriented R/Ts can be interfaced to the PGC with a minimum of external circuitry. See Figure 1 for a typical system configuration.

This NMOS LSI circuit is TTL compatible, operates from a single + 5V supply and is contained in a 16 pin dual in line package.

Table 1. Glossary

TERM/ABBREVIATION	DEFINITION
BCC	Block check character
BTC	Block terminating character
SC	Search character
SSC	Second search character (preceded by DLE)
CRC-16	$X^{16} + X^{15} + X^2 + 1$ divisor, dividend pre-cleared
CRC-12	$X^{12} + X^{11} + X^3 + X^2 + X + 1$ divisor, dividend pre-cleared
LRC-8	Horizontal parity on least significant 7 bits; vertical parity on most significant bit
VRC	Vertical redundancy check (character parity)
R/T	Receiver/transmitter circuit. Also known as USART/USRT/UART/PCI/MPCC
BISYNC	IBM binary synchronous communications (BSC), ANSI X3.28, ISO 1745
MSB	Most significant bit
LSB	Least significant bit
Rx	Receive
Tx	Transmit

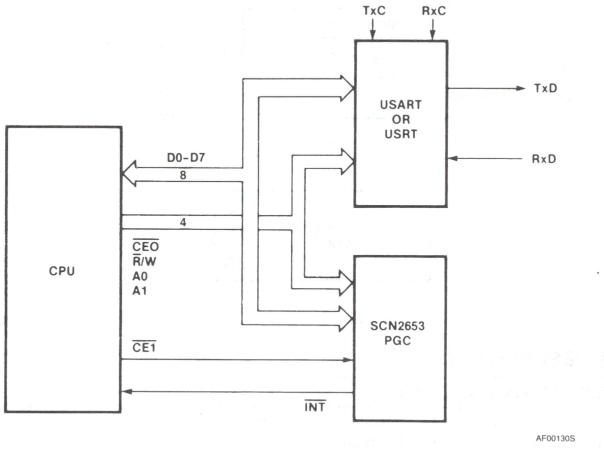


Figure 1. Typical System Configuration

BLOCK DIAGRAM

The PGC consists of the following six major sections: the operation control, character class array, DLE ROM, character register, BCC and parity generators, and BCC registers. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the CPU data bus via a data bus buffer.

Operation Control Unit

This functional block stores configuration and operation instructions from the CPU and generates appropriate signals to control the device operation. It also contains read and write circuits to permit communications between the CPU and the PGC registers via the data bus. The mode, command, and status registers are in this logic block.

Character Register

Characters to be considered for BCC generation, parity generation and checking, or character comparisons are loaded into this register by either $\overline{CE0}$ or $\overline{CE1}$. This register serves as an input to the BCC and VRC generator, where the accumulation and parity generation takes place. The character register also serves as the input for character class array and DLE comparisons.

Character Class Array

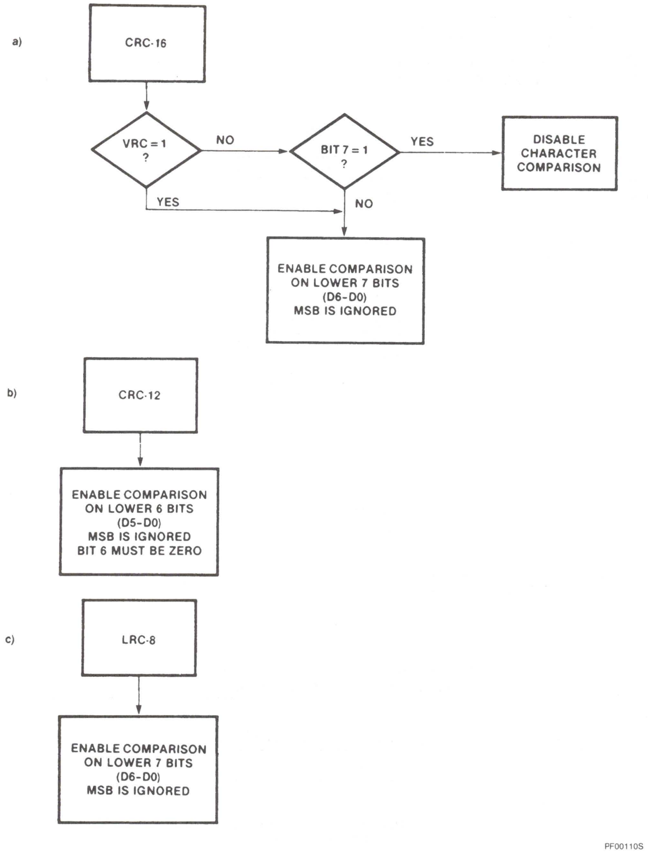
This 128 × 2 array holds the character class associated with each of 128 possible 7-bit characters. The array is zero after a master reset. When the character class array is loaded (see PGC Addressing), the character on the data bus is placed in the class specified by the contents of command register bits CR2 and CR3. The PGC uses these two command bits to represent four different character classes. These are:

1. Normal class (included in the accumulation)
2. SYN character/BISYNC not included class
3. Block terminating character/search class
4. Second search character class (preceded by DLE)

These encoded character classes are used by the PGC:

1. To control the BCC accumulation of associated characters in BISYNC modes only. BCC accumulation in automatic or single accumulation modes is carried out independent of the character classes.
2. To detect characters and two character sequences in all modes of accumulation and to set the control character detect bits in the status register.

It should be noted that any number of characters (up to 128 for CRC-16 or LRC-8; up to 64 for CRC-12) can be put into any one class.


a)
CRC-16
VRC = 1 ?
NO
BIT 7 = 1 ?
YES
DISABLE CHARACTER COMPARISON
YES
NO
ENABLE COMPARISON ON LOWER 7 BITS (D6–D0) MSB IS IGNORED
b)
CRC-12
ENABLE COMPARISON ON LOWER 6 BITS (D5–D0) MSB IS IGNORED BIT 6 MUST BE ZERO
c)
LRC-8
ENABLE COMPARISON ON LOWER 7 BITS (D6–D0) MSB IS IGNORED
PF00110S

NOTE:
Transparent mode always disables the SSC comparison.

Figure 2. Character Class Array Comparison Operation

If VRC is specified along with CRC-16 then the least significant 7 bits of the character are used for character array comparison. If VRC is not enabled, but CRC-16 is, the MSB of the character then determines whether a character comparison is to take place. If the MSB is 0, the comparison takes place; if the MSB is 1, the comparison does not take place and the character is processed as though it were in the normal class. This enables the PGC to detect all communication control characters and DLE-SSC sequences.

Only the first 64 locations of the array are accessed if CRC-12 is selected. The user should right justify each six bit character (D0-D5) to be written into the character class array. Bit 6 must be zero.

If VRC is enabled, the generated parity becomes the most significant bit of the character to be compared. VRC is not allowed in BISYNC transparent mode.

The method in which the character register contents is compared against the character class array depends on the BCC polynomial chosen. Figure 2 illustrates the comparison process.

DLE Read Only Memory

The DLE characters are stored internally and are selected by the error polynomial as follows:

CRC-12: 01 1111

LCR-8 or CRC-16:
No VRC or odd VRC: 0001 0000
Even VRC: 1001 0000

BCC and Parity Generator

This functional block performs all the necessary computation to generate and update the BCC accumulation on a character by character busis. It contains the three generator polynomials (CRC-16, CRC-12, and LRC-8) that can be selected to compute the BCC. This block also checks and generates odd or even parity for 7-bit (ASCII) characters.

BCC Registers

This block consists of two 8-bit registers (BCC upper and BCC lower) which contain the high and low order bytes of the BCC accumulation. The result of the accumulation from the BCC and parity generator is stored in these registers. A recirculating register address pointer is initialized by a power on, master reset, or clear BCC command. The pointer alternately selects BCC upper and lower on successive BCC register accesses for CRC-16 or CRC-12. For LRC-8, BCC upper is always selected.

BCC upper and lower are cleared by a clear BCC or master reset command. The highest term of the BCC polynomial is always represented by bit 0 of BCC upper; the lowest term is always represented by bit 7 of BCC lower (see Figure 3, Orientation of BCC Polynomials.)

The length of the block check character depends on the error checking polynomial that is selected. If LRC-8 is chosen, the BCC result is stored entirely in BCC upper. The BCC lower remains unchanged from previous setting. Both BCC registers are used when CRC-16 is specified. When CRC-12 is selected, the block check character is 12 bits long. The six least significant bits of the BCC are stored in the least significant bits of the BCC lower. The remaining upper six bits of the BCC are stored in least significant bits of BCC upper. The two most significant bits in each BCC register are filled with zero.

The BCC register(s) are read by the CPU after the last data character is transmitted. They can then be sent to the R/T to complete a transmitted block of data. These registers are read and loaded when one PGC is time-shared by several R/Ts. Refer to Applications Information—Multiplexed PGC.

PGC Addressing

All internal registers and the character class array are selected by the unique address codes shown in Table 2.

$$\frac{M(X)}{G(X)} = Q(X) + \frac{R(X)}{G(X)},\ \text{WHERE } R(X) = A_nX^n + A_{n-1}X^{n-1} \cdot \cdot \cdot + A_0X^0$$

M (X) : BINARY POLYNOMIAL (DATA STREAM)
G (X) : FIXED DIVISOR TO GENERATE BCC
Q (X) : QUOTIENT AFTER BCC GENERATION
R (X) : REMAINDER AFTER BCC GENERATION

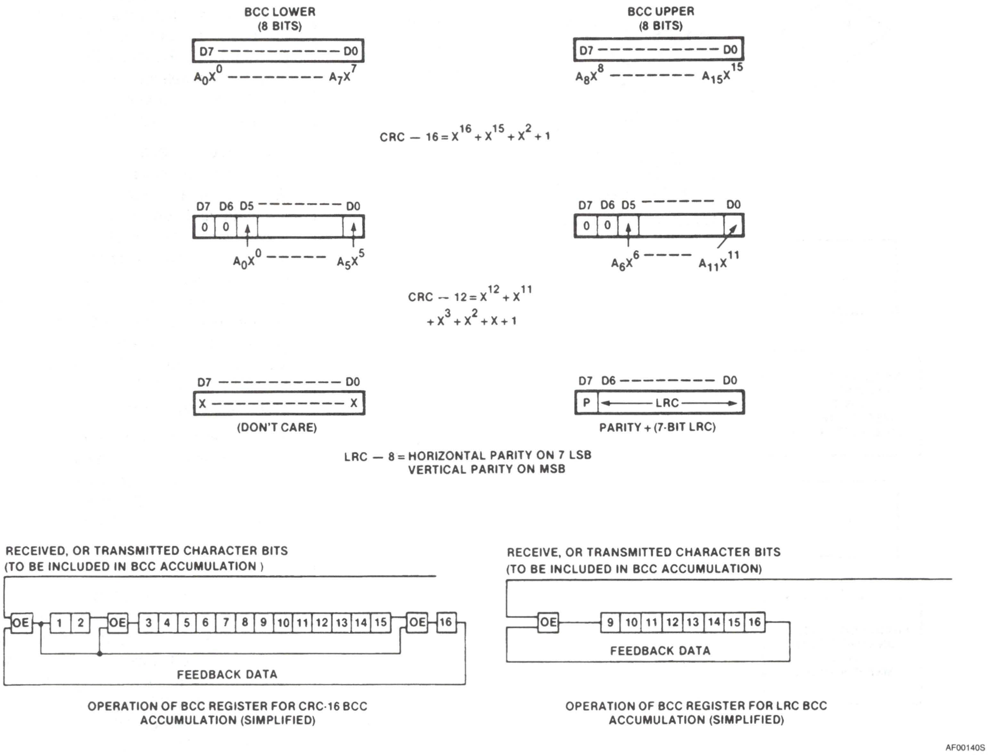

BCC LOWER (8 BITS)
D7 D0
A_0X^0 A_7X^7
BCC UPPER (8 BITS)
D7 D0
A_8X^8 $A_{15}X^{15}$
CRC — 16 = $X^{16} + X^{15} + X^2 + 1$
D7 D6 D5 D0
0 0
A_0X^0 A_5X^5
D7 D6 D5 D0
0 0
A_6X^6 $A_{11}X^{11}$
CRC — 12 = $X^{12} + X^{11} + X^3 + X^2 + X + 1$
D7 D0
X X
(DON'T CARE)
D7 D6 D0
P LRC
PARITY + (7-BIT LRC)
LRC — 8 = HORIZONTAL PARITY ON 7 LSB
VERTICAL PARITY ON MSB
RECEIVED, OR TRANSMITTED CHARACTER BITS
(TO BE INCLUDED IN BCC ACCUMULATION)
OE 1 2 OE 3 4 5 6 7 8 9 10 11 12 13 14 15 OE 16
FEEDBACK DATA
OPERATION OF BCC REGISTER FOR CRC-16 BCC ACCUMULATION (SIMPLIFIED)
RECEIVE, OR TRANSMITTED CHARACTER BITS
(TO BE INCLUDED IN BCC ACCUMULATION)
OE 9 10 11 12 13 14 15 16
FEEDBACK DATA
OPERATION OF BCC REGISTER FOR LRC BCC ACCUMULATION (SIMPLIFIED)
AF00140S

Figure 3. Orientation of BCC Polynomials

Table 2. Address Codes

$\overline{\mathrm{CE0}}$	$\overline{\mathrm{CE1}}$	A1	A0	$\overline{\mathrm{R}}$/W	FUNCTION
0	0	X	X	X	Operation not guaranteed
0	1	0	0	0	If MR2 = 0 load data bus into character register If MR2 = 1 PGC not selected[1]
0	1	0	0	1	If MR2 = 1 load data bus into character register If MR2 = 0 PGC not selected[1]
0	1	0	1	X	PGC not selected[1]
0	1	1	0	X	PGC not selected[1]
0	1	1	1	X	PGC not selected[1]
1	0	0	0	0	Read character register
1	0	0	0	1	Load data bus into character register if MR1,0 ≠00[2]; write character class array using CR3, CR2 class code if MR 1,0 = 00[3, 4]
1	0	0	1	0	Read Status register
1	0	0	1	1	Write command register
1	0	1	0	0	Read mode register
1	0	1	0	1	Write mode register
1	0	1	1	0	Read BCC upper/lower[5]
1	0	1	1	1	Write BCC upper/lower[5]
1	1	X	X	X	PGC not selected[1]

NOTES:

1. Data bus is 3-State.
2. Character will not be accumulated unless MR3 = 1.
3. Character will not be accumulated even if MR3 = 1.
4. The mode bits MR1 and MR0 are cleared to 00 by power-on-reset, master reset, or by loading the mode register bits MR1 and MR0.
5. Recirculating internal pointer selects BCC upper on first access, BCC lower on next access for all BCCs except for LRC-8; in case of LRC-8, the pointer only selects BCC upper.

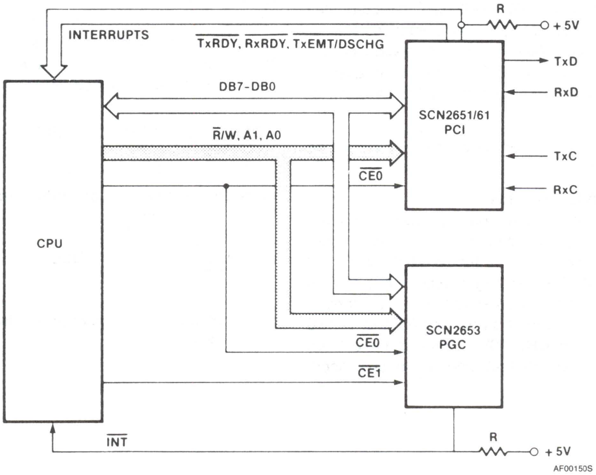

INTERRUPTS
$\overline{\mathrm{TxRDY}}$, $\overline{\mathrm{RxRDY}}$, $\overline{\mathrm{TxEMT/DSCHG}}$
R
+5V
TxD
RxD
DB7-DB0
SCN2651/61
PCI
$\overline{\mathrm{R}}$/W, A1, A0
TxC
RxC
$\overline{\mathrm{CE0}}$
CPU
SCN2653
PGC
$\overline{\mathrm{CE0}}$
$\overline{\mathrm{CE1}}$
$\overline{\mathrm{INT}}$
R
+5V
AF00150S

Figure 4. SCN2651 or SCN2661/2653 Interface

INTERFACE SIGNALS AND TIMING

PGC data transfers are controlled by A1, A0, and $\overline{\mathrm{R}}$/W which must be stable prior to the active low going chip enable pulse. $\overline{\mathrm{CE0}}$ is used for PGC monitoring of data transfers between a CPU/DMA controller and a R/T; $\overline{\mathrm{CE1}}$ is used for direct CPU-to-PGC transfers. MR3 must be set prior to loading the character register in order to accumulate or compare characters via $\overline{\mathrm{CE1}}$. The active low (leading) edge of chip enable initiates a PGC read/write cycle; the rising (trailing) edge ends the cycle and also serves as a write strobe.

When loading the character, mode, or command register, the data bus is strobed into the selected register on the trailing (rising) edge of the appropriate $\overline{\mathrm{CE}}$. When writing into the character class array, the data on the bus (the special character) is placed in the class specified by command register bits CR3 and CR2.

Characters are transferred into the character register when $\overline{\mathrm{CE0}}$ is active (low) depending on the state of MR2 and the $\overline{\mathrm{R}}$/W input. Characters (from the R/T are loaded into the character register when in receive mode (MR2 = 0 and $\overline{\mathrm{R}}$/W = 0) while CPU/DMA characters are loaded into the character register when in transmit mode (MR2 = 1 and $\overline{\mathrm{R}}$/W = 1). The time between consecutive chip enables is given by t_{CEC} or t_{CED}.

The open drain active low interrupt signal ($\overline{\mathrm{INT}}$) goes active whenever one or more of four maskable status conditions (SR0-SR3) are true (= 1). A status read deactivates $\overline{\mathrm{INT}}$.

The same techniques used in interfacing the SCN2651 PCI to 8-bit microprocessors can be used to interface the SCN2653 PGC (consult Application Note M22). Note that when addressing the R/T's holding registers, the PGC pins must have A1, A0 = 00 and that the address and $\overline{\mathrm{R}}$/W signals must be stable (set up) prior to the active low chip enable. When using the SCN2651 or SCN2661 as the R/T, the PGC's A1, A0, $\overline{\mathrm{R}}$/W, and $\overline{\mathrm{CE0}}$ are directly connected to comparable SCN2651 or SCN2661 signals. Schematics of an SCN2653 monitoring data transfers to/from the Signetics SCN2651/2661 and SCN2652 are shown in Figures 4 and 5.

An alternate interfacing technique is to treat the PGC as an independent peripheral device. This necessitates a write character register instruction after the CPU reads or writes a character to or from the R/T.

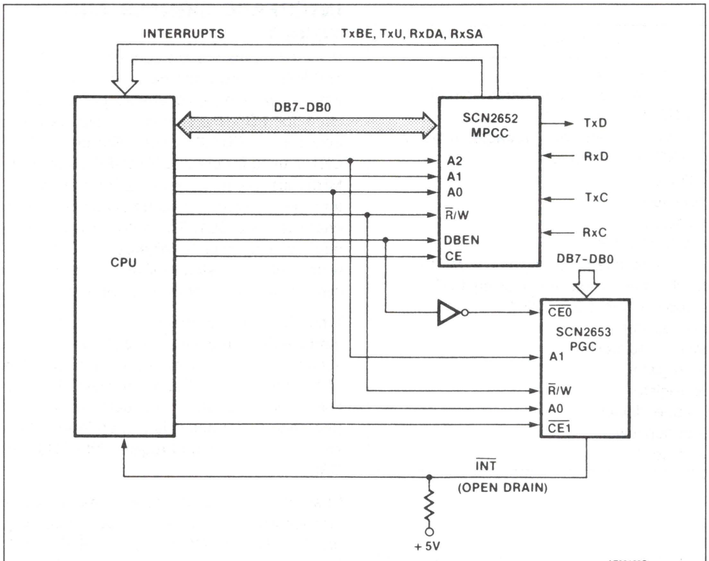


Figure 5. SCN2652/2653 Interface – Typical Protocols: *BISYNC, DDCMP, SDLC, HDLC

PGC PROGRAMMING

The PGC operational mode must be initially programmed by the CPU (see Figure 6). The mode register, command register and character class array should be written into, after a power-on-reset or a master reset command. The character class array should be programmed only for the classes pertinent to the application. After a master reset, the character class array is zero which places all characters in the normal class (included in the BCC accumulation).

OPERATION

The PGC should be initially configured by the CPU (via $\overline{\text{CE1}}$) prior to systems operation. This is done by loading the mode register, command register and character class array (see PGC PROGRAMMING). Characters may then be loaded into the character register for BCC accumulation, VRC generation/checking, BTC/SC and DLE-SSC comparisons. See Table 3 for a summary of BCC accumulation modes.

BCC accumulation depends on the mode selected.

BISYNC Normal

In BISYNC normal mode, all characters loaded into the character register are accumulated except those in the SYN/BISYNC not included class. During receive (MR2 = 0), a BTC/SC match will cause the BCC accumulation to stop after the next one (LRC-8) or two (CRC-12 or CRC-16) characters have been accumulated. At that time, if the BCC accumulation does not equal zero, the BCC error bit (SR0) will be set and $\overline{\text{INT}}$ will go active if the corresponding mask bit (CR4) is enabled (= 1). In transmit (MR2 = 1), the BCC accumulation is automatically stopped once the BTC/SC character has been accumulated. The CPU must read the BCC upper and BCC lower (CRC-12 or CRC-16) register(s) and transmit them to the R/T.

Note that the received BCCs are not subject to VRC if CRC-16 is selected. If LRC-8 is selected, the received BCC is subject to VRC. An incorrect result will set the VRC error bit (SR1). After its accumulation, the least significant 7 bits of BCC upper are checked and a non-zero result will set the BCC error bit (SR0). BCCs are not checked against the character class array nor are they compared to the DLE ROM.

Second search character (SSC) detection is enabled so that a DLE-STX or two character communication control sequence can be detected.

BISYNC Transparent

BISYNC transparent mode should be used for data blocks beginning with DLE-STX if the DLEs are transferred between CPU and R/T ($\overline{\text{CE0}}$) or CPU and PGC ($\overline{\text{CE1}}$), i.e., DLEs are not stripped. VRC should be disabled in this mode. Characters excluded from the BCC accumulation are the first DLE of a DLE-non SYN sequence pair and the DLE-SYN sequence if not preceded by an odd number of DLEs. For example, consider the following transparent mode character string:

DLE	$\underbrace{\textbf{DLE SYN}}$	**DLE**	**DLE**
↑	↑	↗	↗
exclude	**include both**	**exclude**	**include**

$\underbrace{\textbf{DLE SYN}}$	**DLE**	**ETX**
↑	↑	↑
exclude both	**exclude**	**include**

Table 3. Summary of BCC Accumulation Modes

ACCUMULATION MODES	START ACCUMULATION	STOP ACCUMULATION	CHARACTERS EXCLUDED FROM ACCUMULATION
BISYNC normal and BISYNC transparent	Clear BCC registers command Mode register is loaded with BISYNC or automatic mode Start accumulation command Load BCC registers	After BTC has been detected and received BCC is accumulated After transmitted BTC has been accumulated Single mode is selected	SYN/BISYNC not included class in normal mode DLE-SYN/not included class and first DLE of a DLE non SYN pair in transparent mode. These characters are not excluded if preceded by an odd number of DLEs
Automatic	Same as above	Single mode selected	None
Single	Start accumulation command	After each character has been accumulated	Up to user who must generate start accumulation command for each character to be included

In receive and transmit modes, the termination of BCC accumulation works exactly as in BISYNC normal, except that the BTC/SC must be immediately preceded by an odd number of DLEs to be identified as a BTC/SC.

Second search character detection is not enabled in BISYNC transparent.

After a BTC/SC class character is detected by the PGC when receiving in either BISYNC mode, the following one or two characters are accumulated (depending on LRC-8 or CRC-12/16, respectively) and the PGC will automatically stop further accumulation. However, the PGC can continue the accumulation if a start accumulate command is issued or either BISYNC mode is loaded into the mode register. The start accumulate command should be given to the PGC before loading the character that follows the detected BTC/SC. This procedure enables a special search character to be detected (the BTC/SC detect bit (SR2) will be set and an interrupt generated if CR6 = 1) with the BCC accumulation continuing (see Figures 7 and 8).

Automatic Accumulate

All characters loaded into the character register are accumulated, BTC/SC and SSC detection is enabled. The BCC accumulation is not automatically terminated. (The CPU must use single accumulate mode to stop the accumulation). When in receive mode, the BCC error bit (SR0) is set/reset after accumulating each character so that the CPU must examine this bit after the last character is accumulated. SR0 = 0 if the accumulated remainder in the BCC register(s) is zero; otherwise SR0 = 1. Examples of use of automatic accumulate mode usage include an R/T (SCN2651/SCN2661) in transparent DLE/SYN strip mode and asynchronous/synchronous/parallel DDCMP.

Single Accumulate

All characters for which a start accumulate command (CR1, CR0 = 01) is given are accumulated and compared against the character class array. If not given, the BCC accumulation is not updated and BTC/SC and SCC detection is disabled. Operation in this mode is otherwise identical to automatic accumulate.

Single accumulate mode can be used to selectively accumulate characters under CPU control or to accumulate characters that were unintentionally excluded in one of the other modes.

Polynomial Selection and DLE Comparison

The BCC polynomial may be CRC-16, CRC-12 or LRC-8. The cyclic redundancy check (CRC) is generated by dividing the binary value of a character in the character register by the selected polynomial. The quotient is discarded and the remainder is used as the BCC (two 6-bit characters for CRC-12, two 8-bit characters for CRC-16). CRC-16 uses all 8 bits of each BCC register. CRC-12 uses the least significant 6 bits of the BCC registers. The two most significant bits of the BCC registers are cleared to zero whenever CRC-12 is selected (see Figure 3).

When the PGC is in receive mode (MR2 = 0), the received BCC will be accumulated. The result will be zero for an error free message.

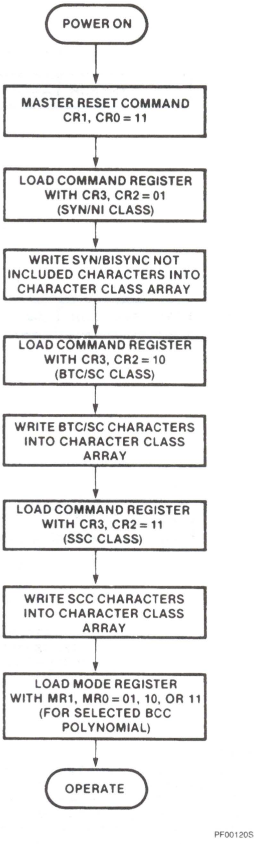
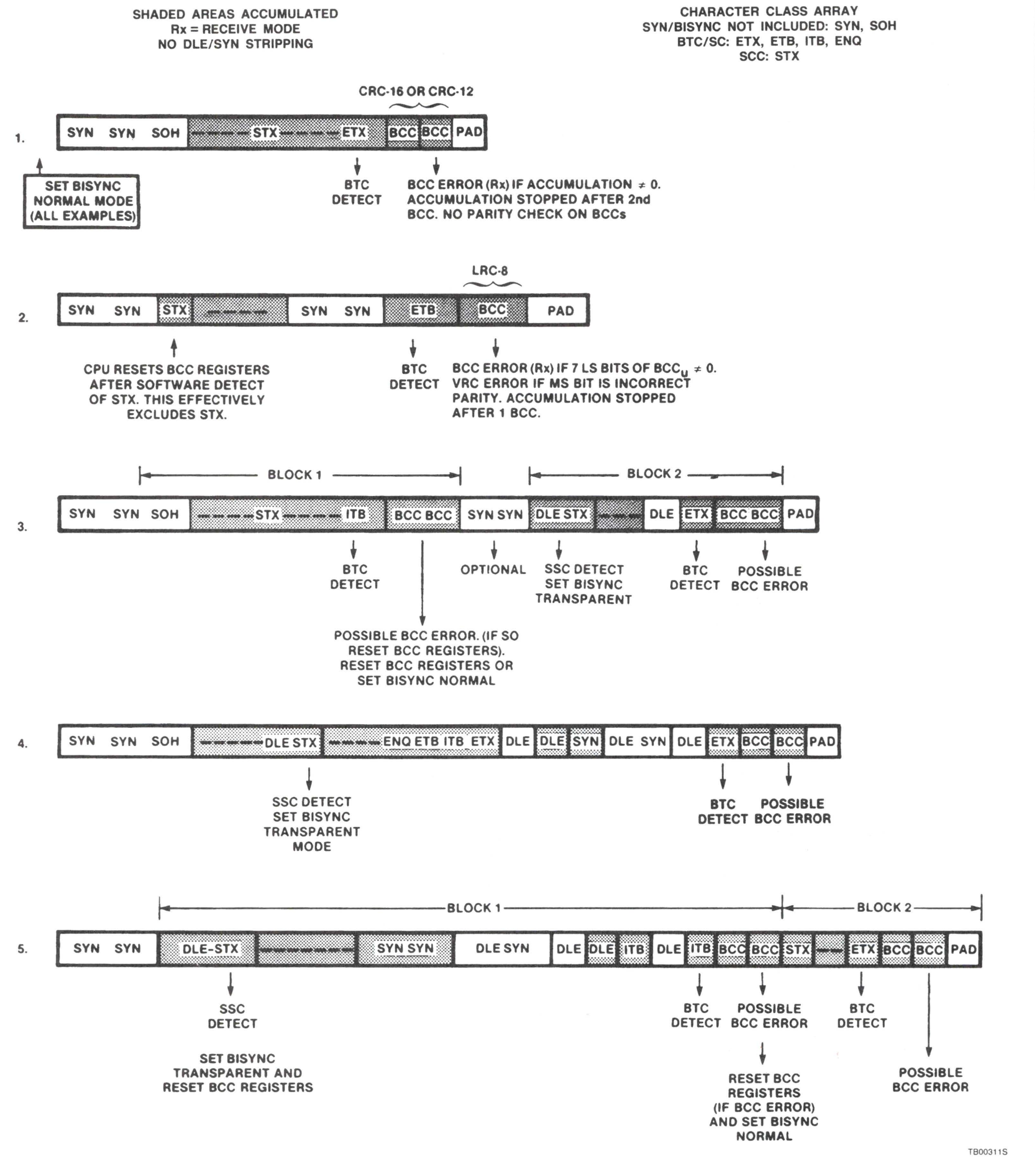
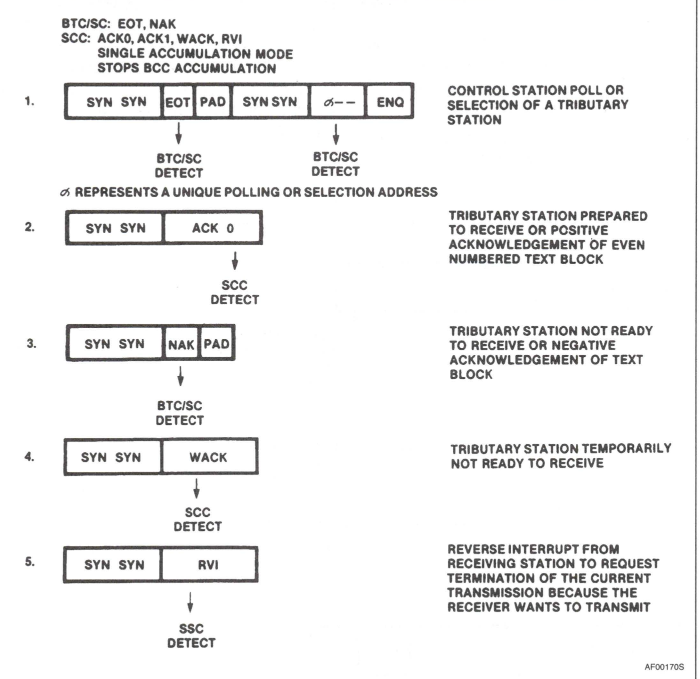

POWER ON
MASTER RESET COMMAND CR1, CR0 = 11
LOAD COMMAND REGISTER WITH CR3, CR2 = 01 (SYN/NI CLASS)
WRITE SYN/BISYNC NOT INCLUDED CHARACTERS INTO CHARACTER CLASS ARRAY
LOAD COMMAND REGISTER WITH CR3, CR2 = 10 (BTC/SC CLASS)
WRITE BTC/SC CHARACTERS INTO CHARACTER CLASS ARRAY
LOAD COMMAND REGISTER WITH CR3, CR2 = 11 (SSC CLASS)
WRITE SCC CHARACTERS INTO CHARACTER CLASS ARRAY
LOAD MODE REGISTER WITH MR1, MR0 = 01, 10, OR 11 (FOR SELECTED BCC POLYNOMIAL)
OPERATE
PF00120S

Figure 6. PGC Programming


SHADED AREAS ACCUMULATED
Rx = RECEIVE MODE
NO DLE/SYN STRIPPING
CHARACTER CLASS ARRAY
SYN/BISYNC NOT INCLUDED: SYN, SOH
BTC/SC: ETX, ETB, ITB, ENQ
SCC: STX
CRC-16 OR CRC-12
1.
SYN SYN SOH STX ETX BCC BCC PAD
SET BISYNC NORMAL MODE (ALL EXAMPLES)
BTC DETECT
BCC ERROR (Rx) IF ACCUMULATION ≠ 0. ACCUMULATION STOPPED AFTER 2nd BCC. NO PARITY CHECK ON BCCs
LRC-8
2.
SYN SYN STX SYN SYN ETB BCC PAD
CPU RESETS BCC REGISTERS AFTER SOFTWARE DETECT OF STX. THIS EFFECTIVELY EXCLUDES STX.
BTC DETECT
BCC ERROR (Rx) IF 7 LS BITS OF BCC$_u$ ≠ 0. VRC ERROR IF MS BIT IS INCORRECT PARITY. ACCUMULATION STOPPED AFTER 1 BCC.
BLOCK 1
BLOCK 2
3.
SYN SYN SOH STX ITB BCC BCC SYN SYN DLE STX DLE ETX BCC BCC PAD
BTC DETECT
OPTIONAL
SSC DETECT SET BISYNC TRANSPARENT
BTC DETECT
POSSIBLE BCC ERROR
POSSIBLE BCC ERROR. (IF SO RESET BCC REGISTERS). RESET BCC REGISTERS OR SET BISYNC NORMAL
4.
SYN SYN SOH DLE STX ENQ ETB ITB ETX DLE DLE SYN DLE SYN DLE ETX BCC BCC PAD
SSC DETECT SET BISYNC TRANSPARENT MODE
BTC DETECT
POSSIBLE BCC ERROR
BLOCK 1
BLOCK 2
5.
SYN SYN DLE-STX SYN SYN DLE SYN DLE DLE ITB DLE ITB BCC BCC STX ETX BCC BCC PAD
SSC DETECT
SET BISYNC TRANSPARENT AND RESET BCC REGISTERS
BTC DETECT
POSSIBLE BCC ERROR
BTC DETECT
RESET BCC REGISTERS (IF BCC ERROR) AND SET BISYNC NORMAL
POSSIBLE BCC ERROR
TB00311S

NOTES:
1. BCC error only for receive mode. In transmit mode, CPU must respond to BTC detect by reading the BCC register(s) and sending them to the R/T. The accumulation is stopped after the BTC is accumulated.
2. ENQ (DLE – ENQ) in a text message should be treated as an abort.
3. Opening SYNs may be stripped by the R/T.
4. The single accumulate mode and command can be used to accumulate a character that inadvertently was excluded. (For example, the DLE of a DLE – STX if the PGC was in transparent mode and there was not a line turnaround prior to the DLE). The single accumulation should be done using $\overline{\text{CE1}}$ after the BCC(s) have been accumulated.

Figure 7. Examples – Bisync Text Message BCC Accumulation


BTC/SC: EOT, NAK
SCC: ACK0, ACK1, WACK, RVI
SINGLE ACCUMULATION MODE
STOPS BCC ACCUMULATION
1.
SYN SYN
EOT
PAD
SYN SYN
ø--
ENQ
BTC/SC DETECT
BTC/SC DETECT
CONTROL STATION POLL OR SELECTION OF A TRIBUTARY STATION
ø REPRESENTS A UNIQUE POLLING OR SELECTION ADDRESS
2.
SYN SYN
ACK 0
SCC DETECT
TRIBUTARY STATION PREPARED TO RECEIVE OR POSITIVE ACKNOWLEDGEMENT OF EVEN NUMBERED TEXT BLOCK
3.
SYN SYN
NAK
PAD
BTC/SC DETECT
TRIBUTARY STATION NOT READY TO RECEIVE OR NEGATIVE ACKNOWLEDGEMENT OF TEXT BLOCK
4.
SYN SYN
WACK
SCC DETECT
TRIBUTARY STATION TEMPORARILY NOT READY TO RECEIVE
5.
SYN SYN
RVI
SSC DETECT
REVERSE INTERRUPT FROM RECEIVING STATION TO REQUEST TERMINATION OF THE CURRENT TRANSMISSION BECAUSE THE RECEIVER WANTS TO TRANSMIT
AF00170S

NOTES:

1. BCC accumulation should be ignored for control messages. This can be effected by single accumulate mode without single accumulate commands.
2. Characters programmed as SSCs should be the binary equivalent of the second character of the DLE – SSC sequence.

Figure 8. Examples – Bisync Control Messages

CRC-12 is used with 6-bit codes. The internal 6-bit transcode DLE character hex 1F is selected by CRC-12. VRC should be disabled (MR4 = 0) for CRC-12 operation. The two most significant bits of the character register are ignored when compared to the internal 6-bit DLE. When the character is checked against the character class array, the MSB is ignored and the next MSB (bit 6) is assumed to be zero. If CRC-12 is specified, the user must write to the character class array with bit 6 cleared.

CRC-16 or LRC-8 implies the use of ASCII or EBCDIC although any 7-bit plus parity or 8-bit no parity code may be used (with DLE = hex 10 or hex 90). The DLE character compare is on an 8-bit basis with the generated parity (if VRC is enabled) as the MSB. When the character is compared against the character class array, the MSB is not used. This may result in a false BTC or SSC detection if there is a VRC error. However, the VRC error bit (SR1) will be set under that condition.

The LRC-8 is generated by the exclusive OR of the 7 least significant bits of the character register and the BCC upper. The most significant bit of the LRC-8 check character is a vertical odd/even parity bit (MR5 = 0/1), which is generated on the least significant bits of that character. The selection of LRC-8 implies VRC is enabled and that only the BCC upper is used for the BCC accumulation. The BCC lower remains unchanged from previous setting.

VRC Generation and Detection

Parity (VRC) is enabled by MR4 and specified as odd or even by MR5. VRC should be disabled when in BISYNC transparent mode and whenever CRC-12 or CRC-16 (EBCDIC) is selected as the BCC polynomial. MR4 = 1 enables VRC generation and detection for both receive and transmit operations. Characters loaded into the character register will have VRC generated on the least significant 7 bits with the generated parity bit written into the character register MSB. If the generated parity does not match the MSB of the loaded character, the VRC error bit (SR1) is set and $\overline{\text{INT}}$ asserted if the corresponding mask bit was enabled (CR5 = 1). Thus, if 7-bit characters are to be transmitted with VRC, CR5 should be zero and SR1 ignored. 8-bit characters with a VRC bit in the MSB position are parity checked by the PGC in both transmit (to R/T) and receive (from R/T) modes, i.e., the PGC operates as a data bus parity checker.

CHARACTER CLASSES

Normal (Included in the Accumulation)

Any character that belongs to this class is normal data, i.e., the character is not a communication control or other special character. Characters in this class are always accumulated in BISYNC, automatic and single accumulation modes.

SYN Character/BISYNC Not Included

SYN characters are never accumulated in BISYNC normal accumulation mode. In BISYNC transparent accumulation mode, the DLE-SYN character pair is not accumulated, but a SYN not preceded by a DLE is accumulated. (DLE is implied as an odd number of DLEs).

Block Termination Character (BTC)/Search Character (SC)

BTC/SC characters have two functions in the PGC: termination of BCC accumulation and character detection. In BISYNC transparent mode, a BTC/SC must be preceded by an odd number of DLEs to be recognized.

Termination of BCC Accumulation

In BISYNC normal and transparent accumulation modes, the PGC will stop the accumulation upon the detection of the BTC/SC character. Examples of BTCs are ETX, ETB, ITB, ENQ.

In receive mode, the accumulation is stopped after the following one (LRC-8) or two (CRC-12, CRC-16) character(s) have been accumulated. In transmit mode, the accumulation is stopped after the BTC/SC character has been accumulated. The BTC/SC character is always accumulated in all of the accumulation modes.

Character Detection

BTC/SC characters will be detected in any of the four accumulation modes when that character is being accumulated. The BTC/SC status bit (SR2) is set on detection. Since detection also stops BISYNC BCC accumulation, the BISYNC accumulation must be restarted if the character is not a BTC. This can be effected by loading BISYNC mode into the mode register or generating a start accumulation command.

Second Search Character Class (SSC)

Control functions in character oriented data link control procedures can be represented by a sequence of two characters, the first character being a DLE. Examples include ACK0, ACK1, WACK, RVI, DISC, WBT and the initiation of transparent text (DLE-STX). The PGC will detect such sequences, except in BISYNC transparent mode, when an SSC class character is being accumulated after being immediately preceded by an odd number of DLEs. Under those conditions, the SSC status bit (SR3) will be set.

The SSC character is always accumulated in all of the accumulation modes.

REGISTER BIT DESCRIPTION

The operation of the PGC is determined by programming the mode register and the command register. The status register provides feedback on potential interrupt conditions. Formats of these registers are shown in Table 4.

Table 4. PGC Register Bit Formats

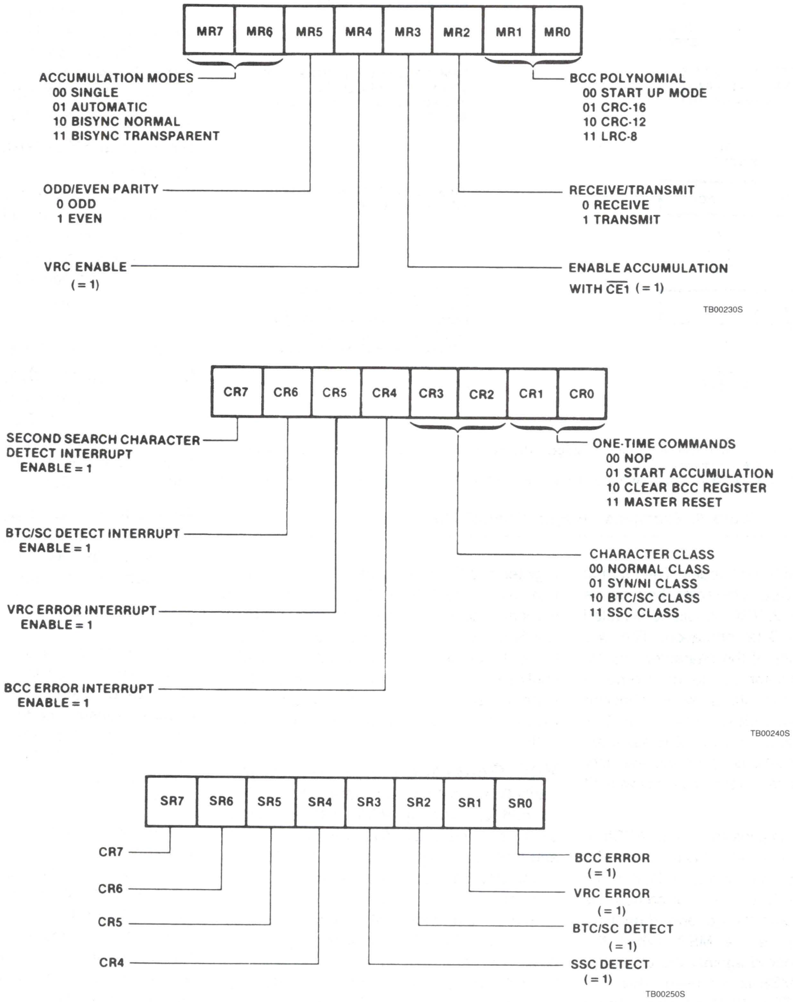

MR7
MR6
MR5
MR4
MR3
MR2
MR1
MR0
ACCUMULATION MODES
00 SINGLE
01 AUTOMATIC
10 BISYNC NORMAL
11 BISYNC TRANSPARENT
BCC POLYNOMIAL
00 START UP MODE
01 CRC-16
10 CRC-12
11 LRC-8
ODD/EVEN PARITY
0 ODD
1 EVEN
RECEIVE/TRANSMIT
0 RECEIVE
1 TRANSMIT
VRC ENABLE
(= 1)
ENABLE ACCUMULATION
WITH $\overline{CE1}$ (= 1)
TB00230S
CR7
CR6
CR5
CR4
CR3
CR2
CR1
CR0
SECOND SEARCH CHARACTER
DETECT INTERRUPT
ENABLE = 1
ONE-TIME COMMANDS
00 NOP
01 START ACCUMULATION
10 CLEAR BCC REGISTER
11 MASTER RESET
BTC/SC DETECT INTERRUPT
ENABLE = 1
CHARACTER CLASS
00 NORMAL CLASS
01 SYN/NI CLASS
10 BTC/SC CLASS
11 SSC CLASS
VRC ERROR INTERRUPT
ENABLE = 1
BCC ERROR INTERRUPT
ENABLE = 1
TB00240S
SR7
SR6
SR5
SR4
SR3
SR2
SR1
SR0
CR7
CR6
CR5
CR4
BCC ERROR
(= 1)
VRC ERROR
(= 1)
BTC/SC DETECT
(= 1)
SSC DETECT
(= 1)
TB00250S

Table 5. BCC Accumulation by Character Class

CR3	CR2	CLASS	BISYNC NORMAL	BISYNC TRANSPARENT	AUTOMATIC ACCUM	SINGLE ACCUM
0	0	Normal	Yes	Yes	Yes	Yes
0	1	SYN/BISYNC not included	No	Yes, unless preceded by an odd number of DLEs	Yes	Yes
1	0	BTC/SC	Yes	Yes	Yes	Yes
1	1	SSC*	Yes	Yes	Yes	Yes

NOTE:
*Preceded by DLE

Mode Register

The mode register defines general PGC operation characteristics. MR1 and MRO = 00 permit the character class array to be programmed. These bits will be zero after a power on or master reset command. After the character class array is programmed, these bits should be set to 01,10, or 11 to select the CRC-16, CRC-12 or LRC-8 polynomials.

MR2(Tx/$\overline{\mathrm{R}}$x) determines whether or not the PGC is to generate (Tx) or generate and check (Rx) the BCC. It is used with $\overline{\mathrm{R}}$/W to determine if the data bus is to be loaded into the character register when $\overline{\mathrm{CE0}}$, $\overline{\mathrm{CE1}}$, A1, A0 = 0100.

If MR2 = 1: 1) the PGC will generate the BCC but will never set the BCC error bit (SR0). 2) If the $\overline{\mathrm{R}}$/W pin is high when $\overline{\mathrm{CE0}}$, $\overline{\mathrm{CE1}}$, A1, A0 = 0100, then the data bus will be loaded into the character register. If $\overline{\mathrm{R}}$/W is low under these conditions, the PGC is not selected.

If MR2 = 0: 1) the PGC will accumulate the BCC and set the BCC error bit (SR0) when appropriate. 2) If the $\overline{\mathrm{R}}$/W pin is low when $\overline{\mathrm{CE0}}$, $\overline{\mathrm{CE1}}$, A1, A0 = 0100, then the data bus will be loaded into the character register. If $\overline{\mathrm{R}}$/W is high under these conditions, the PGC is not selected.

MR3 is a $\overline{\mathrm{CE1}}$ accumulate/compare enable bit. If MR3 = 0, characters loaded into the character register by $\overline{\mathrm{CE1}}$ are not accumulated, checked against the character class array, or compared to the DLE ROM. Parity will be generated and checked if VRC is enabled (MR4 = 1). The primary use of MR3 = 0 is to generate parity on a 7-bit character which is to be transmitted to an R/T. The CPU loads the character register with the 7-bit character and reads the 8-bit VRC generated character via $\overline{\mathrm{CE1}}$. This 8-bit character is then transferred to the R/T via $\overline{\mathrm{CE0}}$. Another application of MR3 = 0 is for a CPU to interleave parity checking on memory data ($\overline{\mathrm{CE1}}$) with on line R/T data transfer ($\overline{\mathrm{CE0}}$).

If MR3 = 1, characters loaded into the character register by $\overline{\mathrm{CE1}}$ will be accumulated (according to the BCC accumulation mode selected) and compared against the character class array and DLE ROM. This bit setting should be used when the CPU/DMA controller sends data characters to be accumulated or compared to the PGC and the R/T is inactive (off line). If the R/T were active, then a DLE or BTC loaded into the character register via $\overline{\mathrm{CE0}}$ would cause incorrect accumulation and character comparisons if the next character was loaded via $\overline{\mathrm{CE1}}$.

Table 6. BTC/SC and SSC Detection Conditions

CLASS	BISYNC NORMAL	BISYNC TRANSPARENT	AUTO ACCUM	SINGLE ACCUM
BTC/SC	Yes	Yes*	Yes	Yes †
SSC	Yes*	No	Yes*	Yes*†

NOTES:
* Only if immediately preceded by an odd number of DLEs.
† Start accumulate command necessary for detection.

MR4 is a VRC enable bit. If MR4 = 1, VRC is enabled as odd/even by MR5. VRC is generated on the 7 LS bits of the character and the MS bit is checked against the generated parity. If not equal, SR1 is set. If MR4 = 0, VRC is not enabled. MR4 = 0 is used for BISYNC transparent mode with ASCII code, and for both BISYNC modes for EBCDIC and SBT.

MR5 is an odd/even VRC bit. If MR5 = 1, the total number of 1 bits in the character including the parity bit is even. If MR5 = 0, the total number of bits is odd. This bit is ignored if MR4 = 0.

MR7, MR6 select the BCC accumulation mode. These modes have been previously discussed in the operation section.

Command Register

The command register contains four interrupt enables, a 2-bit character class code used when programming the character class array, and 2 bits that specify three one time commands and a NOP.

CR1, CR0 = 00 is a NOP. This bit setting is used when changing CR7-CR2 without affecting any of the 3 one time commands.

CR1, CR0 = 01 is a start BCC accumulation command. In single accumulation mode, the character accumulated is the character that is in the character register at the time the command is given. The accumulation stops immediately after the character has been accumulated. If the command is given in either of the BISYNC or automatic accumulation modes, it enables the PGC to accumulate the BCC starting with the next character loaded into the character register. This is a means of restarting a BISYNC normal accumulation after detection of a BTC/SC that is not a valid BTC (example; CR, LF, TAB). In all accumulation modes, a previously detected DLE will not be cancelled by this command.

CR1, CR0 = 10 is a clear BCC registers command. Both BCC registers are cleared along with the associated internal pointer and SR0-SR3. The pointer points to BCC upper. $\overline{\mathrm{INT}}$ is forced high. This command permits BCC accumulation, starting with the next character loaded into the character register in BISYNC or auto modes. Single accumulate mode requires a start BCC accumulation command.

CR1, CR0 = 11 is a master reset command. All internal registers (except the character register), the internal pointer, and the entire character class array are cleared. $\overline{\mathrm{INT}}$ is forced high.

CR3 and CR2 are used for programming the character class array. During a write character class array instruction, the character corresponding to the 7 LS bits of the data bus is placed in the class contained in CR3 and CR2. The encoded character classes control the accumulation of the associated character as shown in Table 5.

Detection operates under the conditions shown in Table 6.

CR7, CR6, CR5, CR4 are interrupt enables that individually enable/disable $\overline{\mathrm{INT}}$ when the corresponding status register condition is true (set). Each bit is set in order to enable $\overline{\mathrm{INT}}$ upon the condition. Each bit is reset to disable $\overline{\mathrm{INT}}$ upon the condition. The state of these bits may be read via the status register (SR7, SR6, SR5, SR4).

The corresponding status bits (SR3, SR2, SR1, SR0) are set independent of the interrupt enables. The bit assignments are:

CR4–BCC error interrupt enable

CR5–VRC error interrupt enable

CR6–BTC/SC detect interrupt enable

CR7–DLE-SSC detect interrupt enable

Status Register

This register reflects the status of the 4 conditions that are potential interrupt ($\overline{\text{INT}}$) sources and the 4 interrupt enables in the command register. A status register read clears SR0, SR1, SR2, SR3 and deactivates $\overline{\text{INT}}$. These bits are also cleared by a master reset or clear BCC command.

SR0 is a BCC error bit. This bit can only be set in receive mode (MR2 = 0). In BISYNC normal and BISYNC transparent modes, SR0 will be set/reset once the accumulation has been stopped by the detection of the BTC/SC character and accumulation of the BCC(s).

In automatic and single accumulate modes, SR0 is set/reset after each character in the character register has been accumulated.

The rules for the detection of a BCC error are:

SR0 = 1	LRC-8: 7 least significant bits of BCC upper ≠ 0 CRC-12, CRC-16: BCC upper or BCC lower ≠ 0
SR0 = 0	LRC-8: 7 least significant bits of BCC upper = 0 CRC-12, CRC-16: BCC upper and BCC lower = 0

SR1 is a VRC error bit. When set, this bit reports a character parity error (on receive or transmit) when parity is enabled (MR4 = 1). Parity is odd/even as specified by MR5. The parity bit will be regenerated in the character register.

SR2 is a BTC/SC detect bit. When set, this bit indicates the character being accumulated is of the BTC/SC class for BISYNC normal, automatic and single accumulate modes. In BISYNC transparent mode, the BTC/SC character being accumulated must be immediately preceded by an odd number of DLEs for this bit to be set.

SR3 is a DLE SSC detect bit. This bit can only be set when in BISYNC normal, auto, or single accumulated modes. When set, it indicates that the character being accumulated is of the SSC class when that character was immediately preceded by an odd number of DLEs.

SR7, SR6, SR5, SR4 are interrupt enables. These 4 bits reflect the state of the interrupt enable command bits CR7, CR6, CR5, and CR4, as follows:

SR4–BCC error

SR5–VRC error

SR6–BTC/SC detect

SR7–SSC detect

APPLICATIONS INFORMATION

Dedicated PGC

The most efficient use of the 2653 is to dedicate one to each R/T for two way alternate (half duplex) operation or two to each R/T for two way simultaneous (full duplex) operation (see Figure 9). The CPU configures each PGC (using $\overline{\text{CE1}}$) by initializing the the mode register, command register, and character class array. Data transfers to or from the R/T can then be on a DMA basis with each receiver holding register ready signal used as a read request (RREQ) and each transmit holding register available signal used as a write request (WREQ) to the DMA controller. The CPU needs only to respond to enable interrupts from each of the PGCs. The individual $\overline{\text{INT}}$ outputs can be wire-OR'd into single CPU interrupt ($\overline{\text{INTRPT}}$) with one pull up resistor. Each PGC in this system has a unique address that is decoded into the respective chip enables.

The CPU or DMA controller could send a block of memory data to the PGC to be error checked without sending that data to the R/T. In that case, $\overline{\text{CE1}}$ is used.

Multiplexed PGC

One PGC may be time-shared among a few R/Ts if the CPU saves and restores the mode register and partial BCC result in the BCC registers. These registers are accessed via $\overline{\text{CE1}}$. There must be a separate save area for each R/T (serial channel) and a channel pointer indicating the last R/T that transferred or received a data character (see Figure 10).

The loading of the BCC registers will clear SR0-SR3 and all previously detected special characters, i.e., DLE, BTC/SC, BCC (BISYNC modes). The BCC accumulation will start again when the next character is loaded into the character register in all accumulation modes except single. That mode requires a start accumulation command.

Figures 11 and 12 represent software flow diagrams for transmit and receive service requests. Note that interrupts from all other R/Ts must be masked during a read or write to the BCC registers so as not to affect the internal BCC address pointer. It is recommended that all R/T interrupts be masked while servicing an interrupt that accesses any PGC register.

BISYNC Operation

Table 7 is a concise listing of SCN2651/SCN2661 operating modes with recommended corresponding SCN2653 BCC accumulation modes.

Character Comparator

The PGC can be used as a programmable data bus character comparator which monitors data bus transfers (CPU ↔ peripheral, CPU ↔ CPU, CPU ↔ memory, memory ↔ peripheral (via DMA)). The user selectively loads the character class array with BTC/SC and SSC characters to be compared. Status bits will be set and an interrupt can be generated upon SC and DLE–SSC detection. A match on one to 128 different characters or DLE–SSC sequences can be programmed.

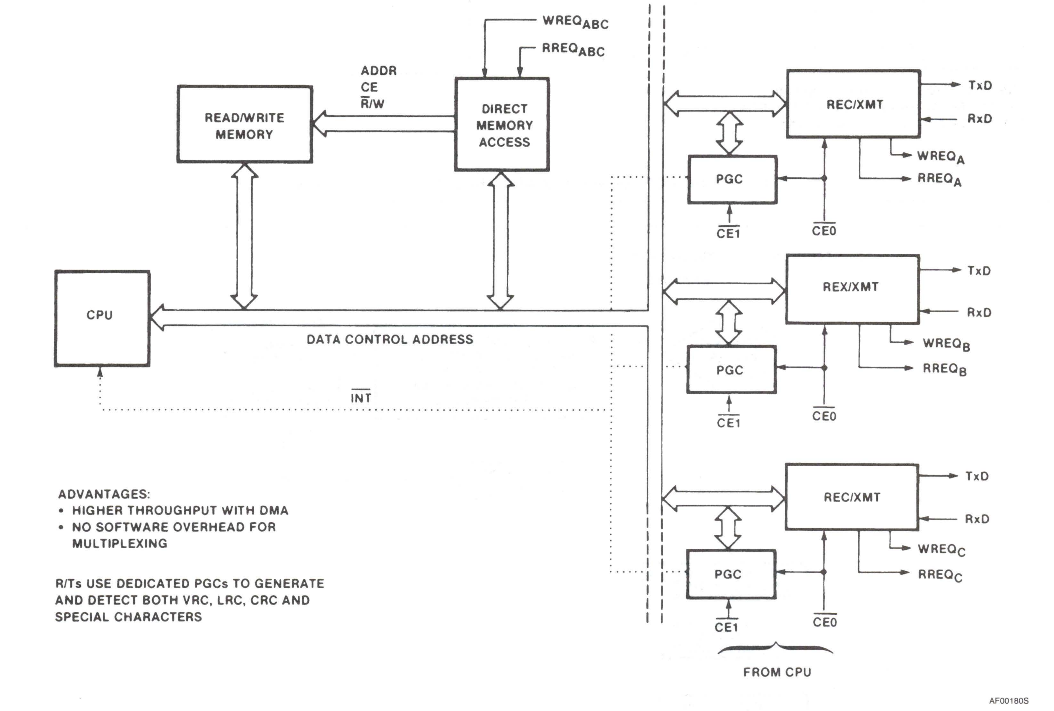
Figure 13 depicts an arrangement where the DMA controller or slave CPU handles data bus transfers, the PGC interrogates the data bus, and the host CPU responds to PGC interrupts.

Polynomial Generator Checker (PGC) SCN2653/SCN68653

Table 7. BISYNC (ANSI 3.28, ISO 1745) Modes for SCN2651/SCN2661 and SCN2653

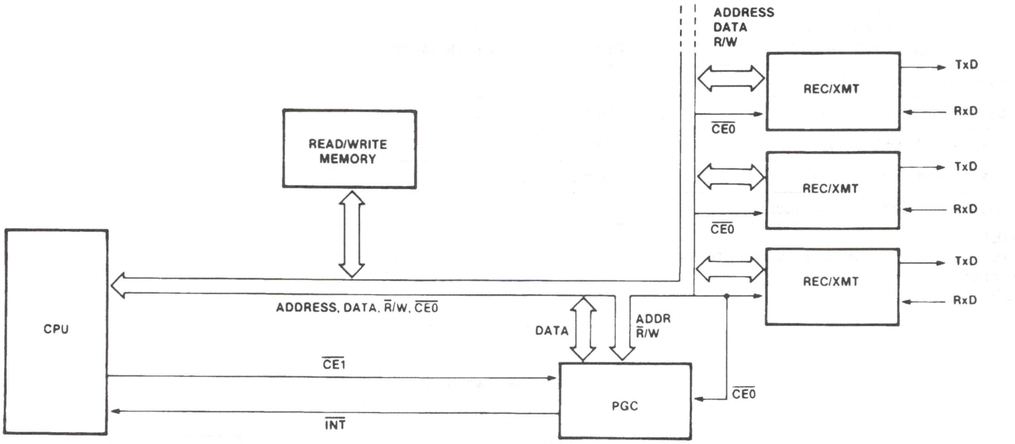
SCN2651/SCN2661 OPERATING MODES	SCN2653 BCC ACCUMULATION MODE
Sync normal non-strip	BISYNC normal
Sync transparent non-strip	BISYNC transparent
Normal SYN/DLE strip[1]	BISYNC normal
Transparent SYN/DLE strip[1]	Automatic accumulate[2]
Async (with SYN/DLE characters)	BISYNC normal

NOTES:
1. CPU should switch to non-strip mode after BTC detect. Otherwise a received BCC could be inadvertently stripped.
2. SSC detect should be ignored.



WREQ$_{ABC}$
RREQ$_{ABC}$
ADDR
CE
R̄/W
READ/WRITE MEMORY
DIRECT MEMORY ACCESS
CPU
DATA CONTROL ADDRESS
INT
REC/XMT
TxD
RxD
WREQ$_A$
RREQ$_A$
PGC
CE1
CE0
REX/XMT
WREQ$_B$
RREQ$_B$
REC/XMT
WREQ$_C$
RREQ$_C$
FROM CPU
ADVANTAGES:
• HIGHER THROUGHPUT WITH DMA
• NO SOFTWARE OVERHEAD FOR MULTIPLEXING
R/Ts USE DEDICATED PGCs TO GENERATE AND DETECT BOTH VRC, LRC, CRC AND SPECIAL CHARACTERS
AF00180S

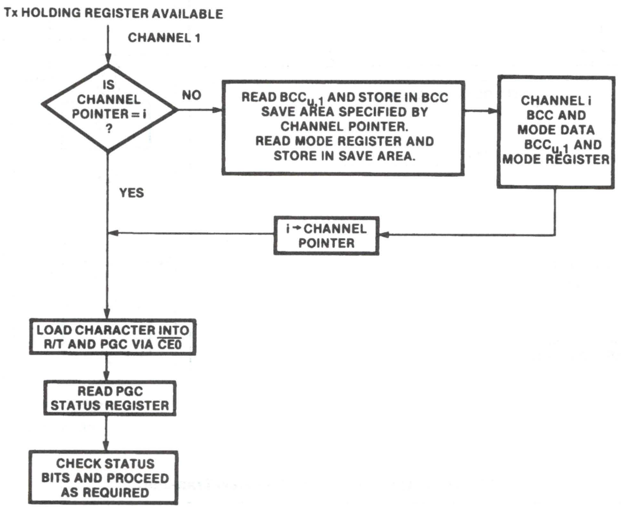
Figure 9. One PGC Per Receiver/Transmitter


ADDRESS
DATA
R/W
REC/XMT
TxD
RxD
$\overline{CE0}$
REC/XMT
TxD
RxD
$\overline{CE0}$
REC/XMT
TxD
RxD
READ/WRITE MEMORY
CPU
ADDRESS, DATA, $\overline{R}$/W, $\overline{CE0}$
DATA
ADDR
$\overline{R}$/W
$\overline{CE1}$
$\overline{INT}$
PGC
$\overline{CE0}$

ONE PGC IS TIME-SHARED BY THREE R/Ts.
THE CPU READS AND RESTORES THE PARTIAL BCC
REMAINDER FOR EACH SERIAL CHANNEL.

AF00190S

Figure 10. PGC Services Multiple Receivers/Transmitters


Tx HOLDING REGISTER AVAILABLE
CHANNEL 1
IS CHANNEL POINTER = i ?
NO
READ $BCC_{u,1}$ AND STORE IN BCC SAVE AREA SPECIFIED BY CHANNEL POINTER. READ MODE REGISTER AND STORE IN SAVE AREA.
CHANNEL i BCC AND MODE DATA $BCC_{u,1}$ AND MODE REGISTER
YES
i → CHANNEL POINTER
LOAD CHARACTER INTO R/T AND PGC VIA $\overline{CE0}$
READ PGC STATUS REGISTER
CHECK STATUS BITS AND PROCEED AS REQUIRED

PF00130S

Figure 11. Multiplexed PGC—Transmit to R/T

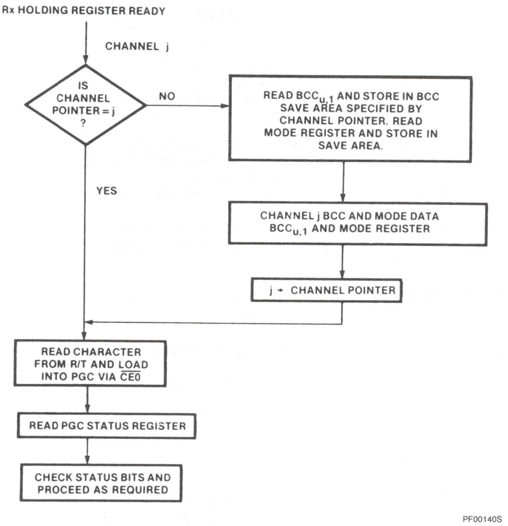
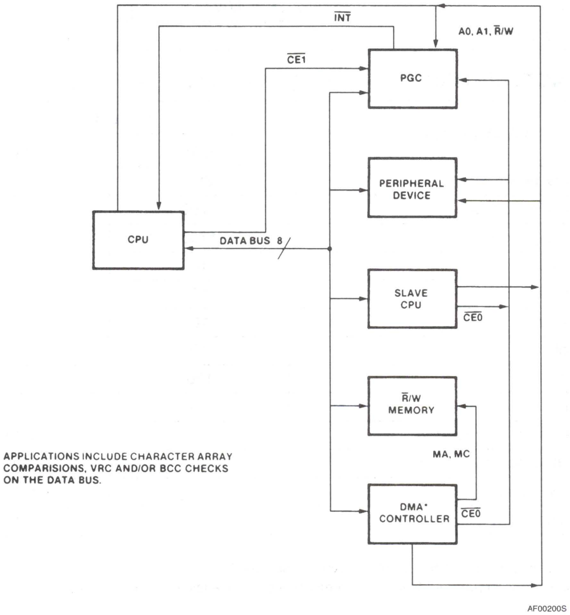

Rx HOLDING REGISTER READY
CHANNEL j
IS CHANNEL POINTER = j ?
NO
YES
READ BCCu,1 AND STORE IN BCC SAVE AREA SPECIFIED BY CHANNEL POINTER. READ MODE REGISTER AND STORE IN SAVE AREA.
CHANNEL j BCC AND MODE DATA BCCu,1 AND MODE REGISTER
j → CHANNEL POINTER
READ CHARACTER FROM R/T AND LOAD INTO PGC VIA CE0
READ PGC STATUS REGISTER
CHECK STATUS BITS AND PROCEED AS REQUIRED
PF00140S

Figure 12. Multiplexed PGC-Receiver From R/T


INT
A0, A1, R/W
CE1
PGC
PERIPHERAL DEVICE
CPU
DATA BUS 8
SLAVE CPU
CE0
R/W MEMORY
MA, MC
DMA* CONTROLLER
CE0
APPLICATIONS INCLUDE CHARACTER ARRAY COMPARISIONS, VRC AND/OR BCC CHECKS ON THE DATA BUS.
AF00200S

NOTE:
* CPU initializes DMA controller for each block transfer of data.

Figure 13. PGC Data Bus Monitoring With DMA Transfers

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	-65 to +150	°C
	All voltages with respect to ground[3]	-0.5 to +6.0	V

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation sections of this specification is not implied.
2. For operating at elevated temperatures the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. However, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 5\%$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Input voltage						
V_{IL} V_{IH}	Low High		 2		0.8	V
Output voltage						
V_{OL} V_{OH}	Low High	$I_{OL} = 2.2mA$ $I_{OH} = -400\mu A$	 2.4	0.25 2.8	0.45	V
I_{IL}	Input load current	$V_{IN} = 0$ to $5.5V$			10	μA
Output leakage current						
I_{LD} I_{LO}	Data bus Open drain	$V_{OUT} = 4.0V$ $V_{OUT} = 4.0V$			10 10	μA
I_{CC}	Power supply current			45	75	mA

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = 5.0V \pm 5\%$[1, 2, 3]

SYMBOL	PARAMETER	LIMITS		UNIT
		Min	Max	
t_{CE}	Chip enable pulse width	250		ns
t_{CED}	Chip enable period D	1750		ns
t_{CEC}[4]	Chip enable period C	1750		ns
t_{AS}	Address setup	10		ns
t_{AH}	Address hold	10		ns
t_{CS}	Control setup	10		ns
t_{CH}	Control hold	10		ns
t_{DS}[5]	Data setup	150		ns
t_{DH}	Data hold	15		ns
t_{DD}[6]	Data delay time for read		200	ns
t_{DF}[6]	Data bus floating time for read		100	ns
t_{INTL}[7]	Interrupt low delay		1600	ns
t_{INTH}[7]	Interrupt high delay		600	ns

NOTES:

1. Parameters are valid over operating temperature range unless otherwise specified.
2. All voltage measurements are referenced to ground. All time measurements are at 50% level for inputs and at the 0.8V or 2V level for outputs. Input levels for testing are 0.45V and 2.4V.
3. Typical values are at +25°C, typical supply voltages and typical processing parameters.
4. $t_{CEC} = 600ns$ during PGC initialization when no BCC accumulation is in progress.
5. $t_{DS} = 50ns$ whenever $\overline{CE0}$ is used.
6. Test conditions: $C_L = 150pF$.
7. $\overline{INT}$ is an open drain output.

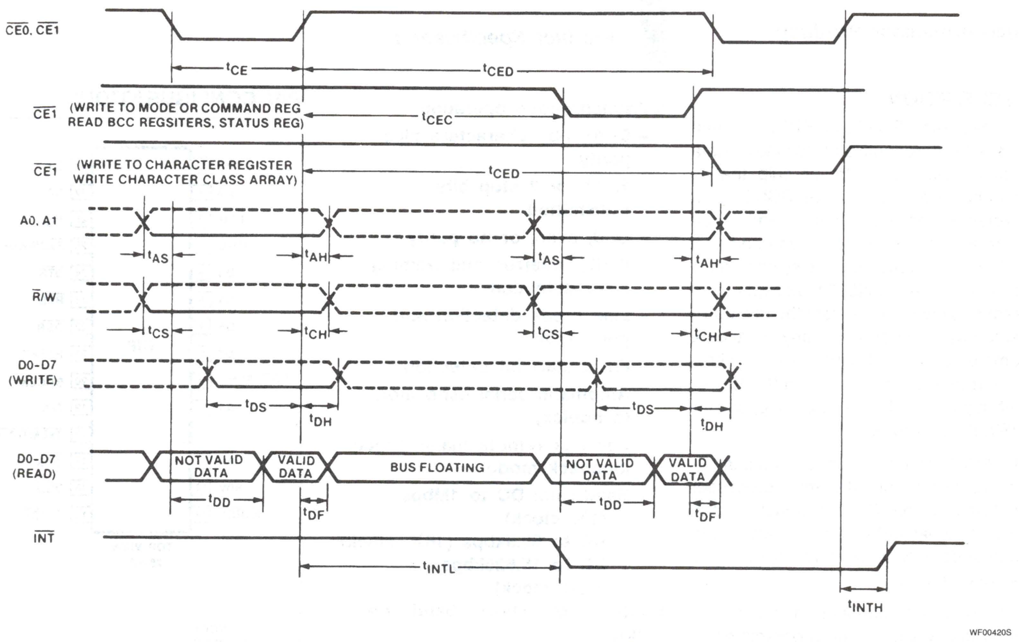

$\overline{CE0}$, $\overline{CE1}$
t_{CE}
t_{CED}
$\overline{CE1}$ (WRITE TO MODE OR COMMAND REG READ BCC REGSITERS, STATUS REG)
t_{CEC}
$\overline{CE1}$ (WRITE TO CHARACTER REGISTER WRITE CHARACTER CLASS ARRAY)
t_{CED}
A0, A1
t_{AS}
t_{AH}
$\overline{R}$/W
t_{CS}
t_{CH}
D0-D7 (WRITE)
t_{DS}
t_{DH}
D0-D7 (READ)
NOT VALID DATA
VALID DATA
BUS FLOATING
t_{DD}
t_{DF}
$\overline{INT}$
t_{INTL}
t_{INTH}
WF00420S

Figure 14. PGC Timing

Signetics

SCN2661/SCN68661

Enhanced Programmable Communications Interface (EPCI)

Microprocessor Products

Product Specification

DESCRIPTION

The Signetics SCN2661 EPCI is a universal synchronous/asynchronous data communications controller chip that is an enhanced version of the SCN2651. It interfaces easily to all 8-bit and 16-bit microprocessors and may be used in a polled or interrupt driven system environment. The SCN2661 accepts programmed instructions from the microprocessor while supporting many serial data communications disciplines — synchronous and asynchronous — in the full- or half-duplex mode. Special support for BISYNC is provided.

The EPCI serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The SCN2661 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode. Each version of the EPCI (A, B, C) has a different set of baud rates.

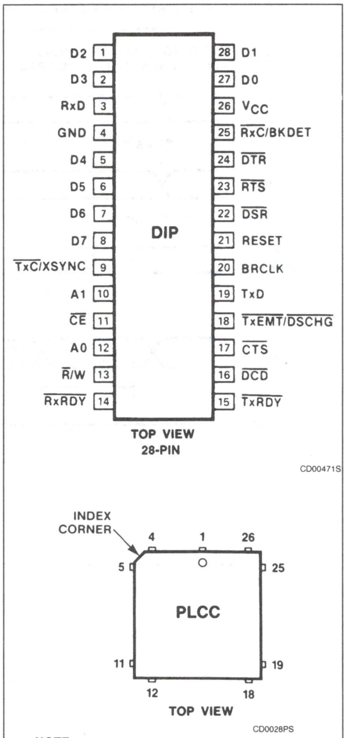
FEATURES

- **Synchronous operation**
  - **5- to 8-bit characters plus parity**
  - **Single or double SYN operation**
  - **Internal or external character synchronization**
  - **Transparent or non-transparent mode**
  - **Transparent mode DLE stuffing (Tx) and detection (Rx)**
  - **Automatic SYN or DLE-SYN insertion SYN, DLE and DLE-SYN stripping**
  - **Odd, even, or no parity**
  - **Local or remote maintenance loopback mode**
  - **Baud rate: DC to 1Mbps (1× clock)**
- **Asynchronous operation**
  - **5- to 8-bit characters plus parity**
  - **1, 1½ or 2 stop bits transmitted**
  - **Odd, even, or no parity**
  - **Parity, overrun and framing error detection**
  - **Line break detection and generation**
  - **False start bit detection**
  - **Automatic serial echo mode (echoplex)**
  - **Local or remote maintenance loopback mode**
  - **Baud rate: DC to 1Mbps (1× clock)**
    DC to 62.5kbps (16× clock)
    DC to 15.625kbps (64× clock)
- **Internal or external baud rate clock**
- **3 baud rate sets**
- **16 internal rates for each set**
- **Double-buffered transmitter and receiver**
- **Dynamic character length switching**
- **Full- or half-duplex operation**
- **TTL compatible inputs and outputs**
- **RxC and TxC pins are short-circuit protected**
- **Single +5V power supply**
- **No system clock required**

APPLICATIONS

- **Intelligent terminals**
- **Network processors**
- **Front-end processors**
- **Remote data concentrators**
- **Computer-to-computer links**
- **Serial peripherals**
- **BISYNC adaptors**

PIN CONFIGURATIONS



NOTE:
Pin Functions the same as 28-pin DIP.

Enhanced Programmable Communications Interface (EPCI)

SCN2661/SCN68661

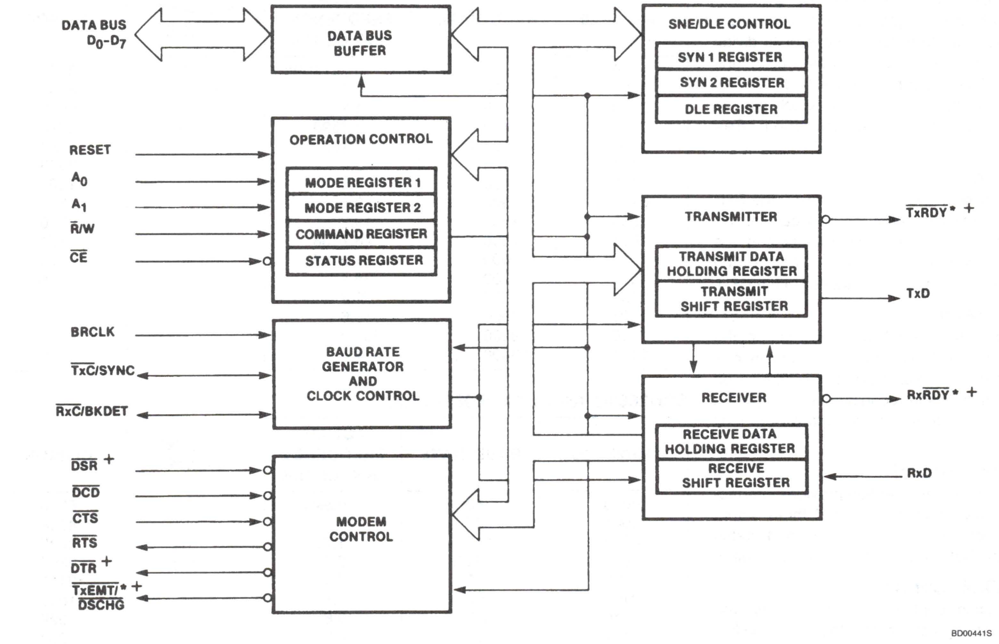
ORDERING INFORMATION

PACKAGES	V_{CC} = +5V ±5%		
	Commercial	Automotive	Military
	0°C to +70°C	-40°C to +85°C	-55°C to +125°C
Ceramic DIP 28-Pin 0.6" Wide	SCN2661AC1I28 SCN2661BC1I28 SCN2661CC1I28	SCN2661AA1I28 SCN2661BA1I28 SCN2661CA1I28	SCN2661AM1I28 SCN2661BM1I28 SCN2661CM1I28
Plastic DIP 28-Pin 0.6" Wide	SCN2661AC1N28 SCN2661BC1N28 SCN2661CC1N28	Contact Factory	Not Available
Plastic LCC	SCN2661AC1A28 SCN2661BC1A28 SCN2661CC1A28	Contact Factory	Not Available

NOTES:

1. See Table 1 for baud rates. Specify SCN2661A, B, or C depending on baud rate selected.
2. The SCN68661 is identical to the SCN2661. Order using part numbers above.

BLOCK DIAGRAM


DATA BUS
D0-D7
DATA BUS BUFFER
SNE/DLE CONTROL
SYN 1 REGISTER
SYN 2 REGISTER
DLE REGISTER
RESET
A0
A1
R/W
CE
OPERATION CONTROL
MODE REGISTER 1
MODE REGISTER 2
COMMAND REGISTER
STATUS REGISTER
TRANSMITTER
TRANSMIT DATA HOLDING REGISTER
TRANSMIT SHIFT REGISTER
TxRDY* +
TxD
BRCLK
TxC/SYNC
RxC/BKDET
BAUD RATE GENERATOR AND CLOCK CONTROL
RECEIVER
RECEIVE DATA HOLDING REGISTER
RECEIVE SHIFT REGISTER
RxRDY* +
RxD
DSR +
DCD
CTS
RTS
DTR +
TxEMT/* + DSCHG
MODEM CONTROL
BD00441S

NOTES:
* Open-drain output pin.
+ 28-pin only.

BLOCK DIAGRAM

The EPCI consists of six major sections: the transmitter; receiver; timing; operation control; modem control, and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers 1 and 2, the command register, and the status register. Details of register addressing and protocol are presented in the EPCI programming section of this data sheet.

Timing

The EPCI contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full-duplex operation (See Table 1).

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

Table 1. Baud Rate Generator Characteristics
SCN2661A (BRCLK = 4.9152MHz)

MR23 – 20	BAUD RATE	ACTUAL FREQUENCY 16× CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	—	6144
0001	75	1.2	—	4096
0010	110	1.7598	-0.01	2793
0011	134.5	2.152	—	2284
0100	150	2.4	—	2048
0101	200	3.2	—	1536
0110	300	4.8	—	1024
0111	600	9.6	—	512
1000	1050	16.8329	0.196	292
1001	1200	19.2	—	256
1010	1800	28.7438	-0.19	171
1011	2000	31.9168	-0.26	154
1100	2400	38.4	—	128
1101	4800	76.8	—	64
1110	9600	153.6	—	32
1111	19200	307.2	—	16

SCN2661B (BRCLK = 4.9152MHz)

MR23 – 20	BAUD RATE	ACTUAL FREQUENCY 16× CLOCK	PERCENT ERROR	DIVISOR
0000	45.5	0.7279kHz	0.005	6752
0001	50	0.8	—	6144
0010	75	1.2	—	4096
0011	110	1.7598	-0.01	2793
0100	134.5	2.152	—	2284
0101	150	2.4	—	2048
0110	300	4.8	—	1024
0111	600	9.6	—	512
1000	1200	19.2	—	256
1001	1800	28.7438	-0.19	171
1010	2000	31.9168	-0.26	154
1011	2400	38.4	—	128
1100	4800	76.8	—	64
1101	9600	153.6	—	32
1110	19200	307.2	—	16
1111	38400	614.4	—	8

SCN2661C (BRCLK = 5.0688MHz)

MR23 – 20	BAUD RATE	ACTUAL FREQUENCY 16× CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	—	6336
0001	75	1.2	—	4224
0010	110	1.76	—	2880
0011	134.5	2.1523	0.016	2355
0100	150	2.4	—	2112
0101	300	4.8	—	1056
0110	600	9.6	—	528
0111	1200	19.2	—	264
1000	1800	28.8	—	176
1001	2000	32.081	0.253	158
1010	2400	38.4	—	132
1011	3600	57.6	—	88
1100	4800	76.8	—	66
1101	7200	115.2	—	44
1110	9600	153.6	—	33
1111	19200	316.8	3.125	16

NOTE:
16× clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1× and BRG can be used only for TxC.

Enhanced Programmable Communications Interface (EPCI)

SCN2661/SCN68661

OPERATION

The functional operation of the SCN2661 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the EPCI programming section of the data sheet.

After programming, the EPCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The SCN2661 is conditioned to receive data when the $\overline{\text{DCD}}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high-to-low (mark to space) transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and one stop bit have been assembled. The data are then transferred to the receive data holding register, the RxRDY bit in the status register is set, and the $\overline{\text{RxRDY}}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the holding register are set to zero. The parity error, framing error, and overrun error status bits are strobed into the status register on the positive going edge of $\overline{\text{RxC}}$ corresponding to the received character boundary. If the stop bit is present, the receiver will immediately begin its search for the next start bit. If the stop bit is absent (framing error), the receiver will interpret a space as a start bit if it persists into the next bit time interval. If a break condition is detected (RxD is low for the entire character as well as the stop bit), only one character consisting of all zeros (with the FE status bit SR5 set) will be transferred to the holding register. The RxD input must return to a high condition before a search for the next start bit begins.

Pin 25 can be programmed to be a break detect output by appropriate setting of MR27 – MR24. If so, a detected break will cause that pin to go high. When RxD returns to mark for one RxC time, pin 25 will go low. Refer to the Break Detection Timing Diagram.

Table 2. CPU-Related Signals

PIN NAME	PIN NO.	INPUT/ OUTPUT	FUNCTION
RESET	21	I	A high on this input performs a master reset on the 2661. This signal asynchronously terminates any device activity and clears the mode, command and status registers. The device assumes the idle state and remains there until initialized with the appropriate control words.
A0, A1	12, 10	I	Address lines used to select internal EPCI registers.
$\overline{\text{R}}$/W	13	I	Read command when low, write command when high.
$\overline{\text{CE}}$	11	I	Chip enable command. When low, indicates that control and data lines to the EPCI are valid and that the operation specified by the $\overline{\text{R}}$/W, A1 and A0 inputs should be performed. When high, places the D0 – D7 lines in the 3-State condition.
D0 – D7	27, 28, 1, 2, 5 – 8	I/O	8-bit, 3-State data bus used to transfer commands, data and status between EPCI and the CPU. D0 is the least significant bit, D7 the most significant bit.
$\overline{\text{TxRDY}}$	15	O	This output is the complement of status register bit SR0. When low, it indicates that the transmit data holding register (THR) is ready to accept a data character from the CPU. It goes high when the data character is loaded. This output is valid only when the transmitter is enabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{\text{RxRDY}}$	14	O	This output is the complement of status register bit SR1. When low, it indicates that the receive data holding register (RHR) has a character ready for input to the CPU. It goes high when the RHR is read by the CPU, and also when the receiver is disabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{\text{TxEMT}}$/ $\overline{\text{DSCHG}}$	18	O	This output is the complement of status register bit SR2. When low, it indicates that the transmitter has completed serialization of the last character loaded by the CPU, or that a change of state of the $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$ inputs has occurred. This output goes high when the status register is read by the CPU, if the TxEMT condition does not exist. Otherwise, the THR must be loaded by the CPU for this line to go high. It is an open drain output which can be used as an interrupt to the CPU. See Status Register (SR2) for details.

When the EPCI is initialized into the synchronous mode, the receiver first enters the hunt mode on a 0 to 1 transition of RxEN (CR2). In this mode, as data are shifted into the receiver shift register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN DETECT status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN DETECT bit to be set. Otherwise, the EPCI returns to the hunt mode (Note that the sequence SYN1 – SYN1 – SYN2 will not achieve synchronization). When synchronization has been achieved, the EPCI continues to assemble characters and transfer them to the holding register, setting the RxRDY status bit and asserting the $\overline{\text{RxRDY}}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN DETECT status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the holding register. Note that the SYN characters used to establish initial synchronization are not transferred to the holding register in any case.

External jam synchronization can be achieved via pin 9 by appropriate setting of MR27 – MR24. When pin 9 is an XSYNC input, the internal SYN1, SYN1 – SYN2, and DLE – SYN1 detection is disabled. Each positive going signal on XSYNC will cause the receiver to establish synchronization on the rising edge of the next RxC pulse. Character assembly will start with the RxD input at this edge. XSYNC may be lowered on the next rising edge of RxC. This external synchronization will cause the SYN DETECT status bit to be set until the status register is read. Refer to XSYNC timing diagram.

Transmitter

The EPCI is conditioned to transmit data when the $\overline{\text{CTS}}$ input is low and the TxEN command register bit is set. The SCN2661 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{\text{TxRDY}}$ output. When the CPU writes a character into the transmit data holding register, these conditions are negated. Data are transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the transmit holding register, the TxD output remains in the marking (high) condition and the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the holding register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the send break command bit (CR3) high.

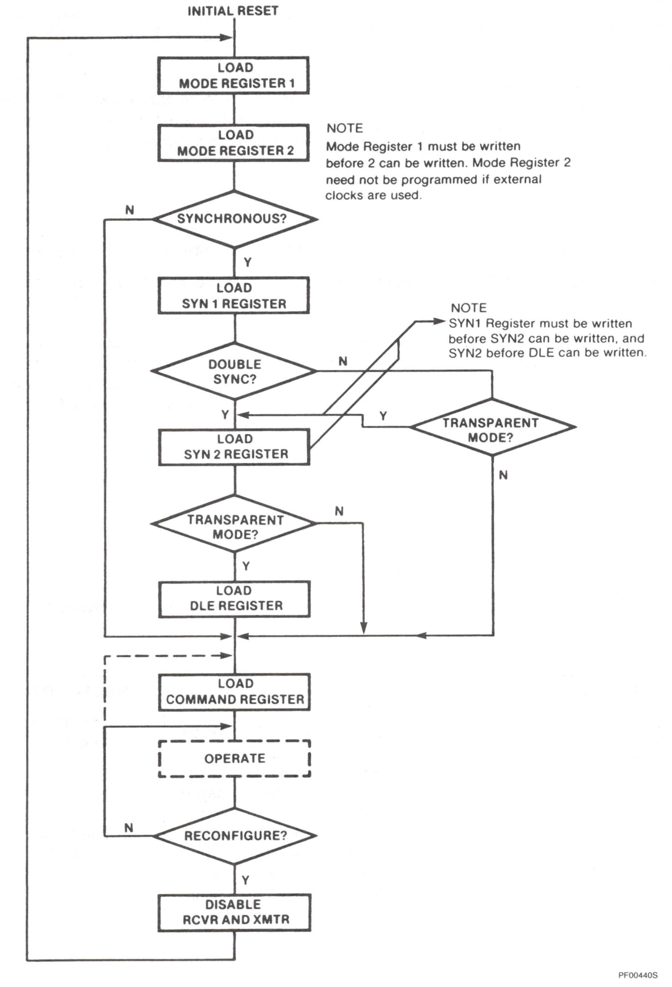
Table 3. Device Related Signals

PIN NAME	PIN NO.	INPUT/ OUTPUT	FUNCTION
BRCLK	20	I	Clock input to the internal baud rate generator (see Table 1). Not required if external receiver and transmitter clocks are used.
$\overline{\text{RxC}}$/BKDET	25	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1×, 16× or 64× the baud rate, as programmed by mode register 1. Data are sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin can be a 1×/16× clock or a break detect output pin.
$\overline{\text{TxC}}$/XSYNC	9	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1×, 16× or 64× the baud rate, as programmed by mode register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, this pin can be a 1×/16× clock output or an external jam synchronization input.
RxD	3	I	Serial data input to the receiver. "Mark" is high, "space" is low.
TxD	19	O	Serial data output from the transmitter. "Mark" is high, "space" is low. Held in mark condition when the transmitter is disabled.
$\overline{\text{DSR}}$	22	I	General purpose input which can be used for data set ready or ring indicator condition. Its complement appears as status register bit SR7. Causes a low output on $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ when its state changes if CR2 or CR0 = 1.
$\overline{\text{DCD}}$	16	I	Data carrier detect input. Must be low in order for the receiver to operate. Its complement appears as status register bit SR6. Causes a low output on $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ when its state changes if CR2 or CR0 = 1. If $\overline{\text{DCD}}$ goes high while receiving, the RxC is internally inhibited.
$\overline{\text{CTS}}$	17	I	Clear to send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the transmit shift register will be transmitted before termination.
$\overline{\text{DTR}}$	24	O	General purpose output which is the complement of command register bit CR1. Normally used to indicate data terminal ready.
$\overline{\text{RTS}}$	23	O	General purpose output which is the complement of command register bit CR5. Normally used to indicate request to send. See Command Register (CR5) for details.

Table 4. SCN2661 Register Addressing

$\overline{CE}$	A_1	A_0	$\overline{R}/W$	FUNCTION
1	X	X	X	3-State data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Write SYN1/SYN2/DLE registers
0	1	0	0	Read mode register ½
0	1	0	1	Write mode register ½
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE:
See AC characteristics section for timing requirements.



NOTE
Mode Register 1 must be written before 2 can be written. Mode Register 2 need not be programmed if external clocks are used.

NOTE
SYN1 Register must be written before SYN2 can be written, and SYN2 before DLE can be written.

Figure 1. SCN2661 Initialization Flowchart

In the synchronous mode, when the SCN2661 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the EPCI unless the CPU fails to send a new character to the EPCI by the time the transmitter has completed sending the previous character. Since synchronous communication does not allow gaps between characters, the EPCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1 – SYN2 doublets, or DLE – SYN1 doublets, depending on the state of MR16 and MR17. Normal transmission of the message resumes when a new character is available in the transmit data holding register. If the send DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character in the THR.

EPCI PROGRAMMING

Prior to initiating data communications, the SCN2661 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The EPCI can be reconfigured at any time during program execution. A flowchart of the initialization process appears in Figure 1.

The internal registers of the EPCI are accessed by applying specific signals to the $\overline{CE}$, $\overline{R}$/W, A1 and A0 inputs. The conditions necessary to address each register are shown in Table 4.

The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions A1 = 0, A0 = 1, and $\overline{R}$/W = 1. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses mode register 1, and a subsequent operation addresses mode register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 register and mode register 1 by a RESET input or by performing a read command register operation, but are unaffected by any other read or write operation.

The SCN2661 register formats are summarized in Tables 5, 6, 7 and 8. Mode registers 1 and 2 define the general operational characteristics of the EPCI, while the command register controls the operation within this basic framework. The EPCI indicates its status in the status register. These registers are cleared when a RESET input is applied.

Mode Register 1 (MR1)

Table 5 illustrates mode register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1× multiplier. 1×, 16×, and 64× multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7 or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asynchronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits. (If 1× baud rate is programmed, 1.5 stop bits defaults to 1 stop bits on transmit.) In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1 – SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE – SYN1 is used for character fill and SYN detect, but the normal synchronization sequence is used to establish character sync. When transmitting, a DLE character in the transmit holding register will cause a second DLE character to be transmitted. This DLE stuffing eliminates the software DLE compare and stuff on each transparent mode data character. If the send DLE command (CR3) is active when a DLE is loaded into THR, only one additional DLE will be transmitted. Also, DLE stripping and DLE detect (with MR14 = 0) are enabled.

The bits in the mode register affecting character assembly and disassembly (MR12 – MR16) can be changed dynamically (during active receive/transmit operation). The character mode register affects both the transmitter and receiver; therefore in synchronous mode, changes should be made only in half-duplex mode (RxEN = 1 or TxEN = 1, but not both simultaneously = 1). In asynchronous mode, character changes should be made when RxEN and TxEN = 0 or when TxEN = 1 and the transmitter is marking in half-duplex mode (RxEN = 0).

To effect assembly/disassembly of the next received/transmitted character, MR12 – 15 must be changed within n bit times of the active going state of $\overline{\text{RxRDY}}/\overline{\text{TxRDY}}$. Transparent and non-transparent mode changes (MR16) must occur within n – 1 bit times of the character to be affected when the receiver or transmitter is active. (n = smaller of the new and old character lengths.)

Table 5. Mode Register 1 (MR1)

MR17	MR16	MR15	MR14	MR13	MR12	MR11	MR10
Sync/Async		**Parity Type**	**Parity Control**	**Character Length**		**Mode and Baud Rate Factor**	
Async: Stop bit length 00 = Invalid 01 = 1 stop bit 10 = 1½ stop bits 11 = 2 stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits		00 = Synchronous 1× rate 01 = Asynchronous 1× rate 10 = Asynchronous 16× rate 11 = Asynchronous 64× rate	
Sync: Number of SYN char 0 = Double SYN 1 = Single SYN	**Sync: Transparency control** 0 = Normal 1 = Transparent						

NOTE:
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16× if internal clock is selected. Mode must be selected (MR11, MR10) in any case.

Table 6. Mode Register 2 (MR2)

MR27 – MR24											**MR23 – MR20**
	TxC	**RxC**	**Pin 9**	**Pin 25**		**TxC**	**RxC**	**Pin 9**	**Pin 25**	**Mode**	**Baud Rate Selection**
0000	E	E	TxC	RxC	1000	E	E	XSYNC^1	RxC/TxC	sync	See baud rates in Table 1
0001	E	I	TxC	1×	1001	E	I	TxC	BKDET	async	
0010	I	E	1×	RxC	1010	I	E	XSYNC^1	RxC	sync	
0011	I	I	1×	1×	1011	I	I	1×	BKDET	async	
0100	E	E	TxC	RxC	1100	E	E	XSYNC^1	RxC/TxC	sync	
0101	E	I	TxC	16×	1101	E	I	TxC	BKDET	async	
0110	I	E	16×	RxC	1110	I	E	XSYNC^1	RxC	sync	
0111	I	I	16×	16×	1111	I	I	16×	BKDET	async	

NOTES:
1. When Pin 9 is programmed as XSYNC input, SYN1, SYN1 – SYN2, and DLE – SYN1 detection is disabled.
E = External clock
I = Internal clock (BRG)
1× and 16× are clock outputs.

Table 7. Command Register (CR)

CR7 CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode	**Request To Send**	**Reset Error**	**Sync/ Async**	**Receive Control (RxEN)**	**Data Terminal Ready**	**Transmit Control (TxEN)**
00 = Normal operation 01 = Async: Automatic echo mode Sync: SYN and/or DLE stripping mode 10 = Local loopback 11 = Remote loopback	0 = Force $\overline{\text{RTS}}$ output high one clock time after TxSR serialization 1 = Force $\overline{\text{RTS}}$ output low	0 = Normal 1 = Reset error flags in status register (FE, OE, PE/DLE detect.)	**Async:** **Force break** 0 = Normal 1 = Force break **Sync** **Send DLE** 0 = Normal 1 = SendDLE	0 = Disable 1 = Enable Not applicable in	0 = Force $\overline{\text{DTR}}$ output high 1 = Force $\overline{\text{DTR}}$ output low	0 = Disable 1 = Enable

Table 8. Status Register (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	**Data Carrier Detect**	**FE/SYN Detect**	**Overrun**	**PE/DLE Detect**	**TxEMT DSCHG**	**RxRDY**	**TxRDY**
0 = $\overline{\text{DSR}}$ input is high 1 = $\overline{\text{DSR}}$ input is low	0 = $\overline{\text{DCD}}$ input is high 1 = $\overline{\text{DCD}}$ input is low	**Async:** 0 = Normal 1 = Framing error **Sync:** 0 = Normal 1 = SYN detected	0 = Normal 1 = Overrun error **Sync:**	**Async:** 0 = Normal 1 = Parity error 0 = Normal 1 = Parity error or DLE received	0 = Normal 1 = Change in $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$, or transmit shift register is empty	0 = Receive holding register empty 1 = Receive holding register has data	0 = Transmit holding register busy 1 = Transmit holding register empty

Mode Register 2 (MR2)

Table 6 illustrates mode register 2. MR23, MR22, MR21 and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable for each EPCI version (−1, −2, −3). Versions 1 and 2 specify a 4.9152MHz TTL input at BRCLK (pin 20); version 3 specifies a 5.0688MHz input which is identical to the Signetics 2651. MR23 – 20 are don't cares if external clocks are selected (MR25 – MR24 = 0). The individual rates are given in Table 1.

MR24 – MR27 select the receive and transmit clock source (either the BRG or an external input) and the function at Pins 9 and 25. Refer to Table 6.

Command Register (CR)

Table 7 illustrates the command register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. A 0-to-1 transition of CR2 forces start bit search (async mode) or hunt mode (sync mode) on the second $\overline{\text{RxC}}$ rising edge. Disabling the receiver causes $\overline{\text{RxRDY}}$ to go high (inactive). If the transmitter is disabled, it will complete the transmission of the character in the transmit shift register (if any) prior to terminating operation. The TxD output will then remain in the marking state (high) while $\overline{\text{TxRDY}}$ and $\overline{\text{TxEMT}}$ will go high (inactive). If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be neglected. A 0 to 1 transition of CR2 will initiate start bit search (async) or hunt mode (sync).

Bits CR1 (DTR) and CR5 (RTS) control the $\overline{\text{DTR}}$ and $\overline{\text{RTS}}$ outputs. Data at the outputs are the logical complement of the register data.

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The TxD line will go high for at least one bit time before beginning transmission of the next character in the transmit data holding register. In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the transmit data holding register. Since this is a one time command, CR3 does not have to be reset by software. CR3 should be set when entering and exiting transparent mode and for all DLE – non-DLE character sequences.

Setting CR4 causes the error flags in the status register (SR3, SR4, and SR5) to be cleared. This is a one time command. There is no internal latch for this bit.

When CR5 (RTS) is set, the $\overline{\text{RTS}}$ pin is forced low. A 1-to-0 transition of CR5 will cause $\overline{\text{RTS}}$ to go high (inactive) one TxC time after the last serial bit has been transmitted. If a 1-to-0 transition of CR5 occurs while data is being transmitted, $\overline{\text{RTS}}$ will remain low (active) until both the THR and the transmit shift register are empty and then go high (inactive) one TxC time later.

The EPCI can operate in one of four submodes within each major mode (synchronous or asynchronous). The operational sub-mode is determined by CR7 and CR6. CR7 – CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the mode and status register instructions.

In asynchronous mode, CR7 – CR6 = 01 places the EPCI in the automatic echo mode. Clocked, regenerated received data are automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU to receiver communications continues normally, but the CPU to transmitter link is disabled. Only the first character of a break condition is echoed.

Table 9. SCN2661 EPCI vs SCN2651 PCI

FEATURE	EPCI	PCI
1. MR2 Bit 6, 7	Control pins 9, 25	Not used
2. DLE detect — SR3	SR3 = 0 for DLE - DLE, DLE - SYN1	SR3 = 1 for DLE - DLE, DLE - SYN1
3. Reset of SR3, DLE detect	Second character after DLE, or receiver disable, or CR4 = 1	Receiver disable, or CR4 = 1
4. Send DLE - CR3	One time command	Reset via CR3 on next $\overline{\mathrm{TxRDY}}$
5. DLE stuffing in transparent mode	Automatic DLE stuffing when DLE is loaded except if CR3 = 1	None
6. SYN1 stripping in double sync non-transparent mode	All SYN1	First SYN1 of pair
7. Baud rate versions	Three	One
8. Terminate ASYNC transmission (drop RTS)	Reset CR5 in response to $\overline{\mathrm{TxEMT}}$ changing from 1 to 0	Reset CR0 when $\overline{\mathrm{TxEMT}}$ goes from 1 to 0. Then reset CR5 when $\overline{\mathrm{TxEMT}}$ goes from 1 to 0
9. Break detect	Pin 25^1	FE and null character
10. Stop bit searched	One	Two
11. External jam sync	Pin 9^2	No
12. Data bus timing	Improved over 2651	—
13. Data bus drivers	Sink 2.2mA Source 400μA	Sink 1.6mA Source 100μA

NOTES:

1. Internal BRG used for RxC.
2. Internal BRG used for TxC.

The TxD output will go high until the next valid start is detected. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. $\overline{\mathrm{TxRDY}}$ output = 1.
4. The $\overline{\mathrm{TxEMT/DSCHG}}$ pin will reflect only the data set change condition.
5. The TxEN command (CR0) is ignored.

In synchronous mode, CR7 - CR6 = 01 places the EPCI in the automatic SYN/DLE stripping mode. The exact action taken depends on the setting of bits MR17 and MR16:

1. In the non-transparent, single SYN mode (MR17 - MR16 = 10), characters in the data stream matching SYN1 are not transferred to the receive data holding register (RHR).
2. In the non-transparent, double SYN mode (MR17 - MR16 = 00), characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR.
3. In transparent mode (MR16 = 1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR. However, only the first DLE of a DLE-DLE pair is stripped.

Note that automatic stripping mode does not affect the setting of the DLE detect and SYN detect status bits (SR3 and SR5).

Two diagnostic sub-modes can also be configured. In local loopback mode (CR7 - CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{\mathrm{DTR}}$ is connected to $\overline{\mathrm{DCD}}$ and $\overline{\mathrm{RTS}}$ is connected to $\overline{\mathrm{CTS}}$.
3. The receiver is clocked by the transmit clock.
4. The $\overline{\mathrm{DTR}}$, $\overline{\mathrm{RTS}}$ and $\overline{\mathrm{TxD}}$ outputs are held high.
5. The $\overline{\mathrm{CTS}}$, $\overline{\mathrm{DCD}}$, $\overline{\mathrm{DSR}}$ and RxD inputs are ignored.

Additional requirements to operate in the local loopback mode are that CR0 (TxEN), CR1 (DTR), and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the EPCI.

The second diagnostic mode is the remote loopback mode (CR7 - CR6 = 11). In this mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. No data are sent to the local CPU, but the error status conditions (PE, FE) are set.
4. The $\overline{\mathrm{RxRDY}}$, $\overline{\mathrm{TxRDY}}$, and $\overline{\mathrm{TxEMT/DSCHG}}$ outputs are held high.
5. CR0 (TxEN) is ignored.
6. All other signals operate normally.

Status Register

The data contained in the status register (as shown in Table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the transmitter ready (TxRDY) status bit. It, and its corresponding output, are valid only when the transmitter is enabled. If equal to 0, it indicates that the transmit data holding register has been loaded by the CPU and the data has not been transferred to the transmit shift register. If set equal to 1, it indicates that the holding register is ready to accept data from the CPU. This bit is initially set when the transmitter is enabled by CR0, unless a character has previously been loaded into the holding register. It is not set when the automatic echo or remote loopback modes are programmed. When this bit is set, the $\overline{\mathrm{TxRDY}}$ output pin is low. In the automatic echo and remote loopback modes, the output is held high.

SR1, the receiver ready (RxRDY) status bit, indicates the condition of the receive data holding register. If set, it indicates that a character has been loaded into the holding register from the receive shift register and is ready to be read by the CPU. If equal to zero, there is no new character in the holding register. This bit is cleared when the CPU reads the receive data holding register or when the receiver is disabled by CR2. When set, the $\overline{\mathrm{RxRDY}}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{\mathrm{DSR}}$ or $\overline{\mathrm{DCD}}$ inputs (when CR2 or CR0 = 1) or that the transmit shift register has completed transmission of a character and no new character has been loaded into the transmit data holding register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the transmit data holding register. The DSCHG condition is enabled when TxEN = 1 or RxEN = 1. It is cleared when the status register is read by the CPU. If the status register is read twice and SR2 = 1 while SR6 and SR7 remain unchanged, then a TxEMT condition exists. When SR2 is set, the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching DLE register was received and the present character is neither SYN1 nor DLE. This bit is cleared when the next character following the above sequence is loaded into RHR, when the receiver is disabled, or by a reset error command, CR4.

The overrun error status bit, SR4, indicates that the previous character loaded into the receive holding register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled or by the reset error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by a stop bit; i.e., only the first stop bit is checked. If RHR = 0 when SR5 = 1, a break condition is present. In synchronous nontransparent mode (MR16 = 0), it indicates receipt of the SYN1 character in single SYN mode or the SYN1 – SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1 – SYN2) and, after synchronization has been achieved, when a DLE – SYN1 pair is received. The bit is reset when the receiver is disabled, when the reset error command is given in asynchronous mode, or when the status register is read by the CPU in the synchronous mode.

SR6 and SR7 reflect the conditions of the $\overline{\text{DCD}}$ and $\overline{\text{DSR}}$ inputs, respectively. A low input sets its corresponding status bit, and a high input clears it.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	Note 4	°C
T_{STG}	Storage temperature range	–65 to +150	°C
	All voltages with respect to ground[3]	–0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
Input voltage						
V_{IL}	Low				0.8	V
V_{IH}	High		2			V
Output voltage						
V_{OL}	Low	I_{OL} = 2.2mA			0.4	V
V_{OH}[7]	High	I_{OH} = –400μA	2.4			V
I_{IL}	Input leakage current	V_{IN} = 0 to 5.5V			10	μA
3-State output leakage current						
I_{LH}	Data bus high	V_O = 4V			10	μA
I_{LL}	Data bus low	V_O = 0.45V			10	μA
I_{CC}	Power supply current				150	mA

CAPACITANCE $T_A = 25°C$, $V_{CC} = 0V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
Capacitance						
C_{IN}	Input	f_C = 1MHz Unmeasured pins tied to ground			20	pF
C_{OUT}	Output				20	pF
$C_{I/O}$	Input/Output				20	pF

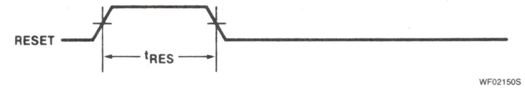
AC ELECTRICAL CHARACTERISTICS[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Pulse width						
t_{RES}	Reset		1000			ns
t_{CE}	Chip enable		250			
Setup and hold time						
t_{AS}	Address setup		10			ns
t_{AH}	Address hold		10			
t_{CS}	$\overline{\text{R}}$/W control setup		10			
t_{CH}	$\overline{\text{R}}$/W control hold		10			
t_{DS}	Data setup for write		150			
t_{DH}	Data hold for write		10			
t_{RXS}	RX data setup		300			
t_{RXH}	RX data hold		350			
t_{DD}	Data delay time for read	$C_L = 150pF$			200	ns
t_{DF}	Data bus floating time for read	$C_L = 150pF$			100	
t_{CED}	CE to CE delay		600			
Input clock frequency						
f_{BRG}	Baud rate generator (2661A, B)		1	4.9152	4.9202	MHz
f_{BRG}	Baud rate generator (2661C)		1	5.0688	5.0738	
$f_{R/T}$[10]	$\overline{\text{TxC}}$ or $\overline{\text{RxC}}$		DC		1	
Clock width						
t_{BRH}[9]	Baud rate high (2661A, B)		75			ns
t_{BRH}[9]	Baud rate high (2661C)		70			
t_{BRL}[9]	Baud rate low (2661A, B)		75			
t_{BRL}[9]	Baud rate low (2661C)		70			
$t_{R/TH}$	$\overline{\text{TxC}}$ or $\overline{\text{RxC}}$ high		480			
$t_{R/TL}$[10]	$\overline{\text{TxC}}$ or $\overline{\text{RxC}}$ low		480			
t_{TXD}	TxD delay from falling edge of $\overline{\text{TxC}}$	$C_L = 150pF$			650	ns
t_{TCS}	Skew between TxD changing and falling edge of $\overline{\text{TxC}}$ output[8]	$C_L = 150pF$		0		

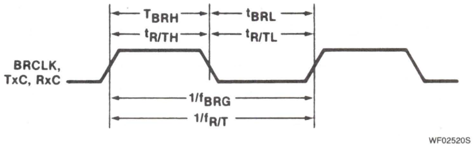
NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified. See ordering code table for applicable temperature range and operating supply range.
5. All voltage measurements are referenced to ground. All time measurements are at the 50% level for inputs (except t_{BRH} and t_{BRL}) and at 0.8V and 2V for outputs. Input levels swing between 0.4V and 2.4V, with a transition time of 20ns maximum.
6. Typical values are at +20°C, typical supply voltages and typical processing parameters.
7. $\overline{\text{INTR}}$, $\overline{\text{TxRDY}}$, $\overline{\text{RxRDY}}$ and $\overline{\text{TxEMT/DSCHG}}$ outputs are open drain.
8. Parameter applies when internal transmitter clock is used.
9. Under test conditions of 5.0688MHz f_{BRG} (2661C) and 4.9152MHz f_{BRG} (2661A,B), t_{BRH} and t_{BRL} measured at V_{IH} and V_{IL}, respectively.
10. In asynchronous local loopback mode, using 1× clock, the following parameters apply: $f_{R/T} = 0.83MHz$ max and $t_{R/TL} = 700ns$ min.

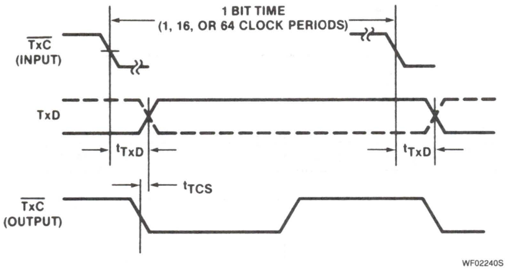
TIMING DIAGRAMS


RESET
tRES
WF02150S

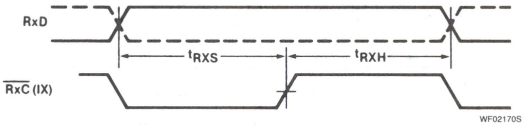
Reset


TBRH
tBRL
tR/TH
tR/TL
BRCLK,
TxC, RxC
1/fBRG
1/fR/T
WF02520S

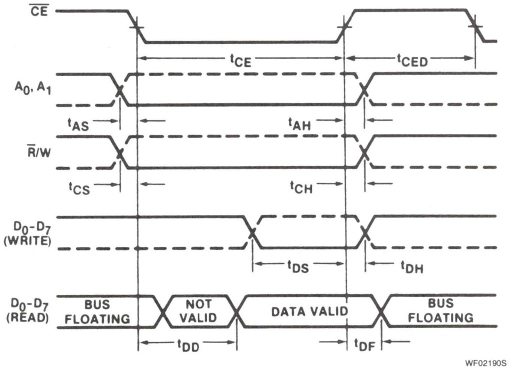
Clock


1 BIT TIME
(1, 16, OR 64 CLOCK PERIODS)
TxC
(INPUT)
TxD
tTxD
tTxD
tTCS
TxC
(OUTPUT)
WF02240S

Transmit

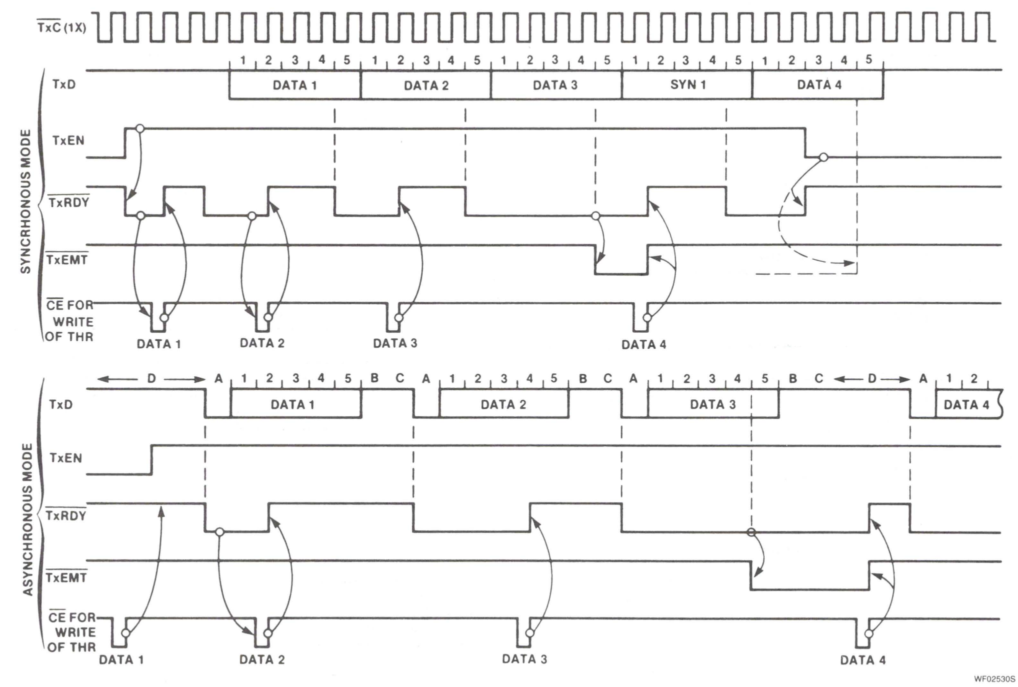

RxD
tRXS
tRXH
RxC (IX)
WF02170S

Receive


CE
tCE
tCED
A0, A1
tAS
tAH
R/W
tCS
tCH
D0-D7
(WRITE)
tDS
tDH
D0-D7
(READ)
BUS
FLOATING
NOT
VALID
DATA VALID
BUS
FLOATING
tDD
tDF
WF02190S

Read and Write

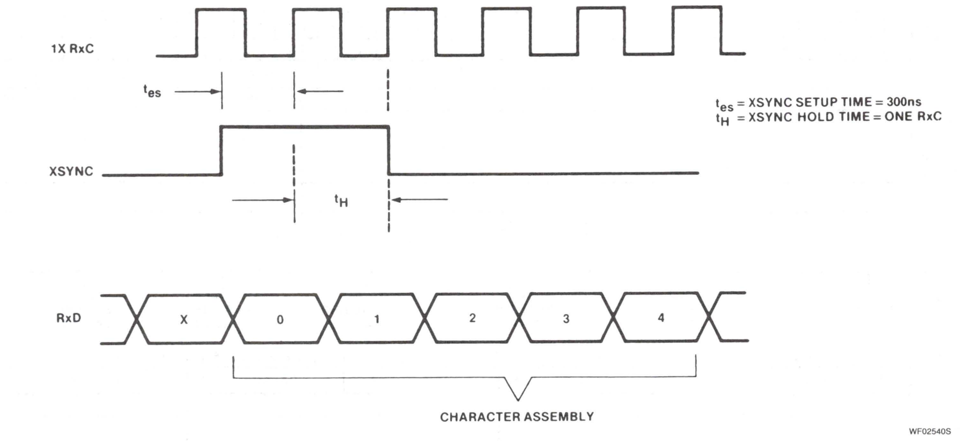
TIMING DIAGRAMS (Continued)


TxC (1X)
SYNCHRONOUS MODE
TxD
TxEN
TxRDY
TxEMT
CE FOR WRITE OF THR
1 2 3 4 5
DATA 1
DATA 2
DATA 3
SYN 1
DATA 4
ASYNCHRONOUS MODE
D
A
B
C
WF02530S

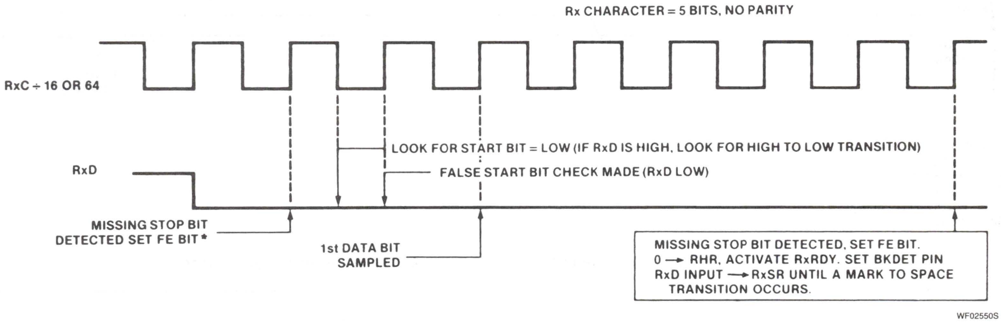
NOTES:
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit.

TxRDY, TxEMT (Shown for 5-bit characters, no parity, 2 stop bits [in asynchronous mode])

TIMING DIAGRAMS (Continued)


1X RxC
t_{es}
t_{es} = XSYNC SETUP TIME = 300ns
t_H = XSYNC HOLD TIME = ONE RxC
XSYNC
t_H
RxD
X
0
1
2
3
4
CHARACTER ASSEMBLY
WF02540S

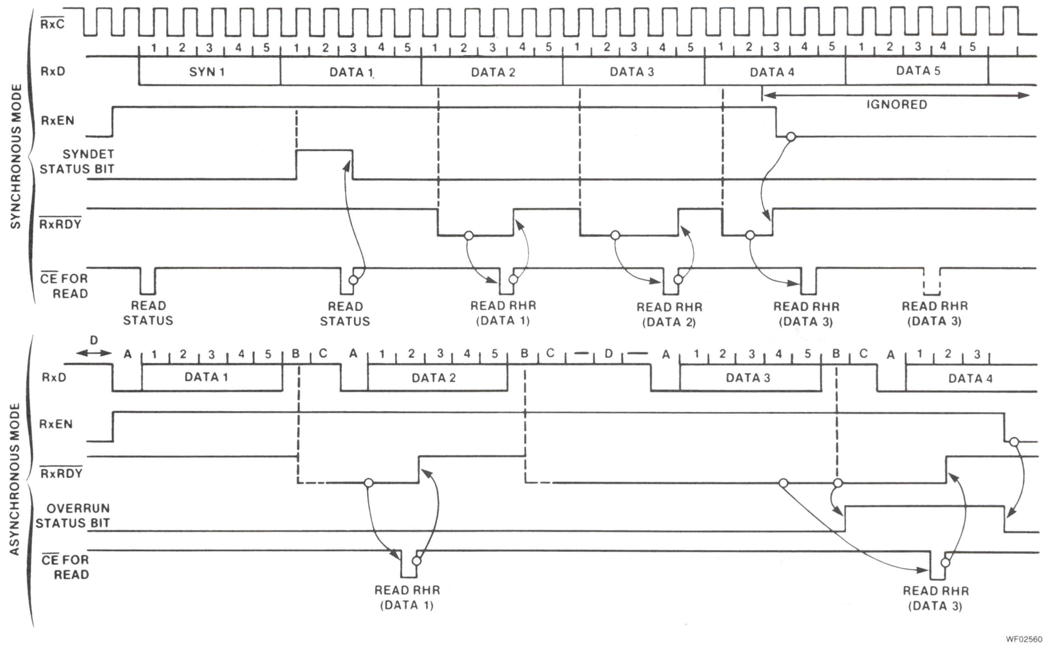
External Synchronization With XSYNC


Rx CHARACTER = 5 BITS, NO PARITY
RxC ÷ 16 OR 64
RxD
LOOK FOR START BIT = LOW (IF RxD IS HIGH, LOOK FOR HIGH TO LOW TRANSITION)
FALSE START BIT CHECK MADE (RxD LOW)
MISSING STOP BIT DETECTED SET FE BIT*
1st DATA BIT SAMPLED
MISSING STOP BIT DETECTED, SET FE BIT.
0 → RHR, ACTIVATE RxRDY. SET BKDET PIN
RxD INPUT → RxSR UNTIL A MARK TO SPACE TRANSITION OCCURS.
WF02550S

NOTE:
* If the stop bit is present, the start bit search will commence immediately.

Break Detection Timing

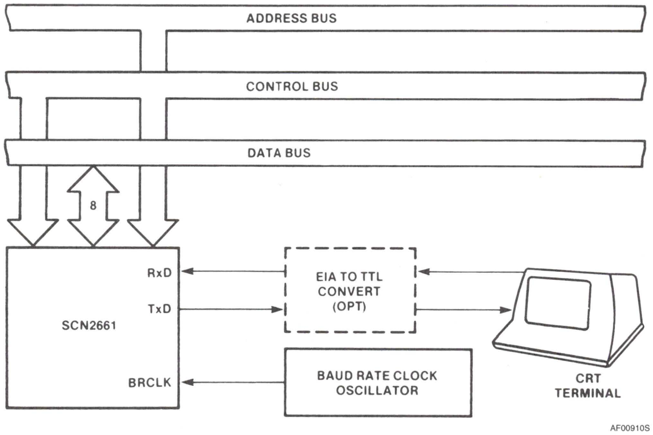
TIMING DIAGRAMS (Continued)


SYNCHRONOUS MODE
RxC
RxD
SYN 1
DATA 1
DATA 2
DATA 3
DATA 4
DATA 5
IGNORED
RxEN
SYNDET STATUS BIT
RxRDY
CE FOR READ
READ STATUS
READ STATUS
READ RHR (DATA 1)
READ RHR (DATA 2)
READ RHR (DATA 3)
READ RHR (DATA 3)
ASYNCHRONOUS MODE
D
A
B
C
RxD
DATA 1
DATA 2
DATA 3
DATA 4
RxEN
RxRDY
OVERRUN STATUS BIT
CE FOR READ
READ RHR (DATA 1)
READ RHR (DATA 3)
WF02560S

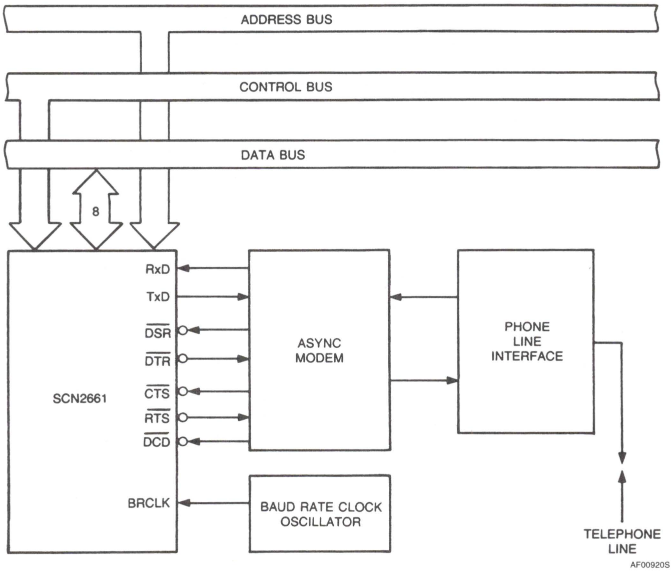
NOTES:
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
Only one stop bit is detected.

RxRDY (Shown for 5-bit characters, no parity, 2 stop bits [in asynchronous mode])

TYPICAL APPLICATIONS

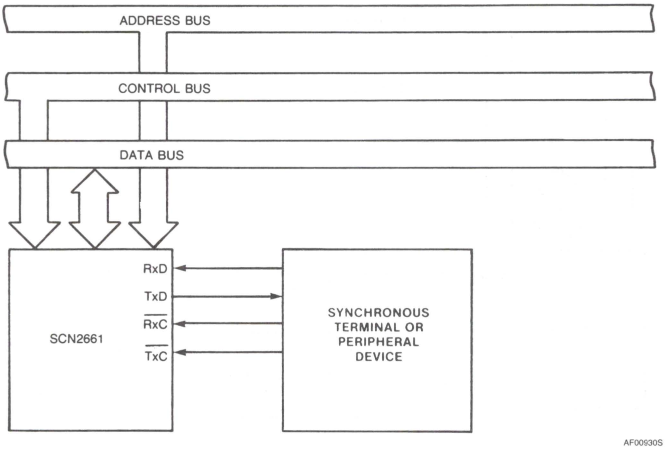

ADDRESS BUS
CONTROL BUS
DATA BUS
8
SCN2661
RxD
TxD
BRCLK
EIA TO TTL CONVERT (OPT)
BAUD RATE CLOCK OSCILLATOR
CRT TERMINAL
AF00910S

Asynchronous Interface to CRT Terminal

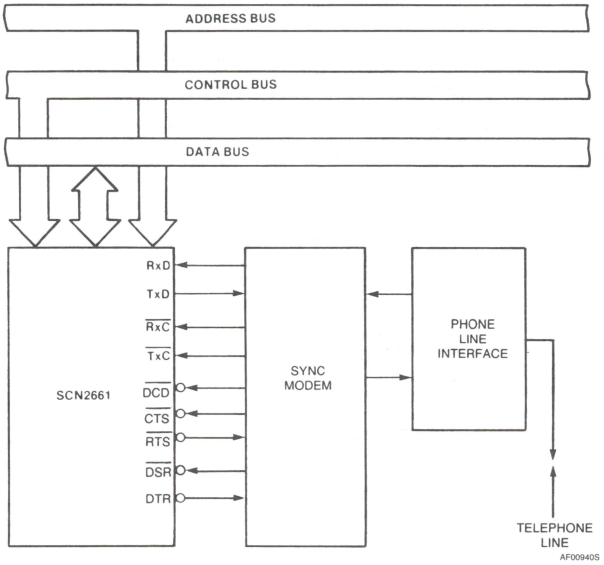

ADDRESS BUS
CONTROL BUS
DATA BUS
8
SCN2661
RxD
TxD
$\overline{DSR}$
$\overline{DTR}$
$\overline{CTS}$
$\overline{RTS}$
$\overline{DCD}$
BRCLK
ASYNC MODEM
PHONE LINE INTERFACE
BAUD RATE CLOCK OSCILLATOR
TELEPHONE LINE
AF00920S

Asynchronous Interface to Telephone Lines

TYPICAL APPLICATIONS (Continued)


ADDRESS BUS
CONTROL BUS
DATA BUS
SCN2661
RxD
TxD
RxC
TxC
SYNCHRONOUS TERMINAL OR PERIPHERAL DEVICE
AF00930S

Synchronous Interface to Terminal or Peripheral Device


ADDRESS BUS
CONTROL BUS
DATA BUS
SCN2661
RxD
TxD
RxC
TxC
DCD
CTS
RTS
DSR
DTR
SYNC MODEM
PHONE LINE INTERFACE
TELEPHONE LINE
AF00940S

Synchronous Interface to Telephone Lines

Signetics

SCN2671

Programmable Keyboard and Communication Controller (PKCC)

Microprocessor Products

Product Specification

DESCRIPTION

The Signetics SCN2671 Programmable Keyboard and Communications Controller (PKCC) is a MOS LSI device which provides a versatile keyboard encoder and an independent full-duplex asynchronous communications controller. It is intended for use in microprocessor based systems and provides an 8-bit data bus interface.

The keyboard encoder handles the scanning, debounce, and encoding of a keyboard matrix with a maximum of 128 keys. It provides four levels of key encoding corresponding to the separate SHIFT and CONTROL input combinations. Four keyboard rollover modes can be programmed including provisions for up to 16 latched keys. Control outputs are provided for interfacing with contact or capacitive keyboards. An 8-bit keyboard status register provides status information to the CPU.

The receiver section of the communications controller accepts serial data from the RxD pin and converts it to parallel data characters. Simultaneously, the transmitter section accepts parallel data from the data bus and outputs serialized data onto the TxD pin. Received data is checked for parity and framing errors, and break conditions are flagged. Character lengths can be programmed as 5, 6, 7, or 8 bits not including parity, start or stop bits. An internal baud rate generator (BRG) with 16 divider ratios can be used to derive the receive and/or transmit clocks. The BRG can accept an external clock or operate directly from a crystal. An 8-bit communications status register provides status information to the CPU.

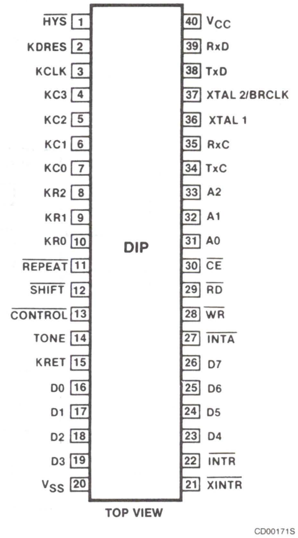
FEATURES

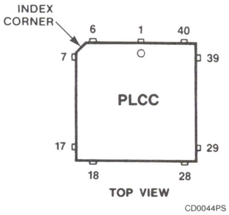
- **Keyboard Interface**
  - **Contact or capacitive keyboard**
  - **Up to 128 keys on an 8 × 16 matrix**
  - **Encoded or unencoded operation**
  - **Four code levels per key**
  - **Latched key option — separate depress and release codes**
  - **Programmable scan rate and debounce time**
  - **Programmable rollover modes**
  - **Programmable auto-repeat for selected keys**
  - **Tone output — two frequencies**
- **Asynchronous communication interface**
  - **Internal baud rate generator — 16 rates**
  - **Full duplex operation**
  - **Detection of start and end of break**
  - **Programmable break generation**
  - **Programmable character parameters**
  - **Auto-echo and maintenance loopback modes**
- **Polled or interrupt operation**
- **Interrupt priority controller and vector generator**
- **Operates directly from crystal or external clocks**
- **TTL compatible**
- **Single +5V power supply**

APPLICATIONS

- **CRT terminals**
- **Hard copy terminals**
- **Word processing systems**
- **Data entry terminals**
- **Small business computers**

PIN CONFIGURATIONS





Pin	Function	Pin	Function
1	NC	23	$\overline{\mathrm{XINTR}}$
2	$\overline{\mathrm{HYS}}$	24	$\overline{\mathrm{INTR}}$
3	KDRES	25	D4
4	KCLK	26	D5
5	KC3	27	D6
6	KC2	28	D7
7	KC1	29	$\overline{\mathrm{INTA}}$
8	KC0	30	$\overline{\mathrm{WR}}$
9	KR2	31	$\overline{\mathrm{RD}}$
10	KR1	32	$\overline{\mathrm{CE}}$
11	KR0	33	A0
12	NC	34	NC
13	$\overline{\mathrm{REPEAT}}$	35	A1
14	$\overline{\mathrm{SHIFT}}$	36	A2
15	$\overline{\mathrm{CONTROL}}$	37	TxC
16	TONE	38	RxC
17	KRET	39	XTAL1
18	D0	40	XTAL2/BRCLK
19	D1	41	TxD
20	D2	42	RxD
21	D3	43	NC
22	V_{SS}	44	V_{CC}

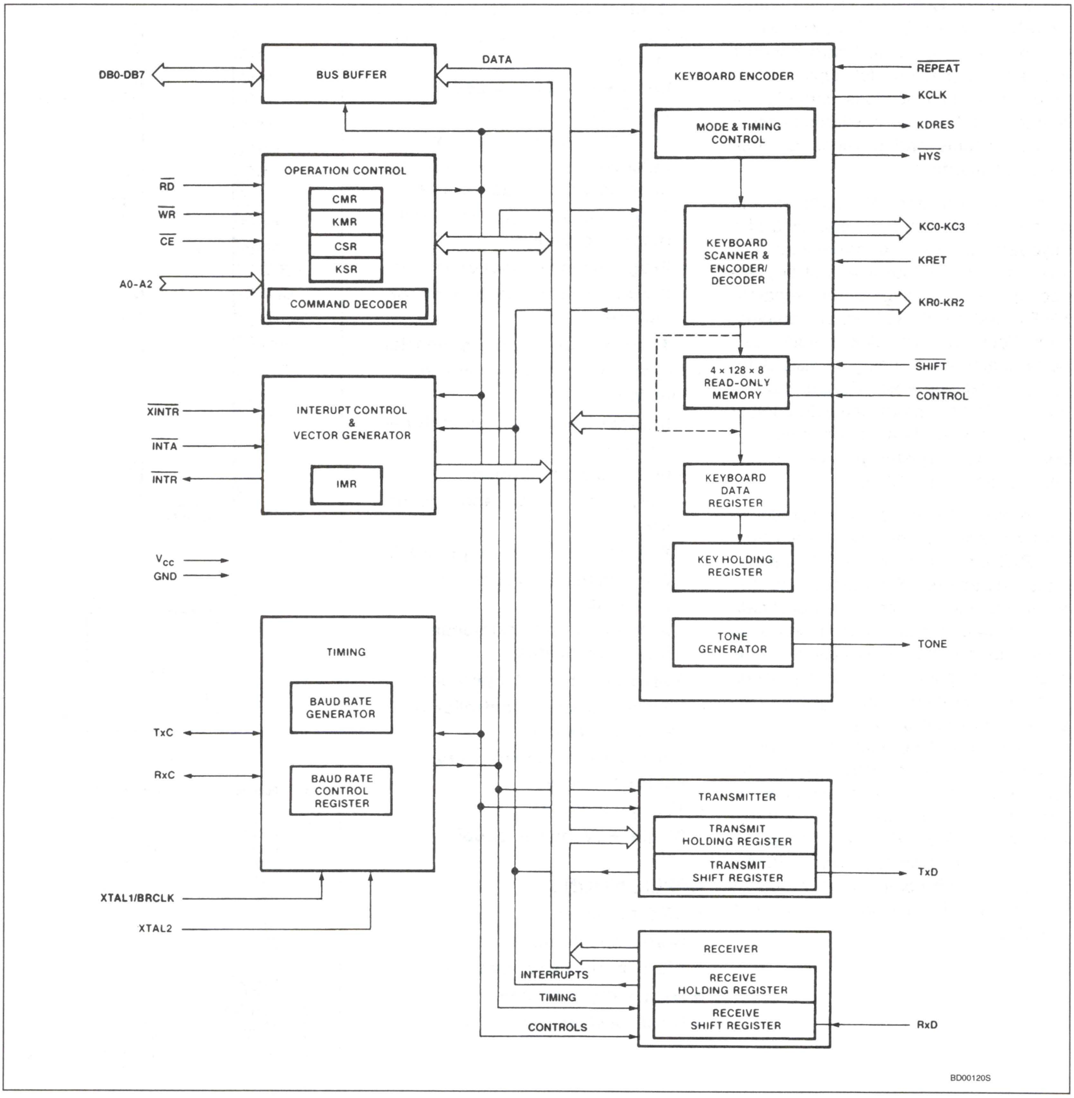
Programmable Keyboard and Communication Controller (PKCC)

SCN2671

ORDERING INFORMATION

PACKAGES	$V_{CC} = +5V \pm 5\%$, $T_A = 0°C$ to $+70°C$
Ceramic DIP	SCN2671AC1I40
Plastic DIP	SCN2671AC1N40
Plastic LCC	SCN2671AC1A44

BLOCK DIAGRAM


DB0-DB7
BUS BUFFER
DATA
KEYBOARD ENCODER
REPEAT
KCLK
KDRES
HYS
MODE & TIMING CONTROL
OPERATION CONTROL
RD
WR
CE
A0-A2
CMR
KMR
CSR
KSR
COMMAND DECODER
KEYBOARD SCANNER & ENCODER/ DECODER
KC0-KC3
KRET
KR0-KR2
4 x 128 x 8 READ-ONLY MEMORY
SHIFT
CONTROL
XINTR
INTA
INTR
INTERUPT CONTROL & VECTOR GENERATOR
IMR
KEYBOARD DATA REGISTER
KEY HOLDING REGISTER
Vcc
GND
TONE GENERATOR
TONE
TIMING
BAUD RATE GENERATOR
BAUD RATE CONTROL REGISTER
TxC
RxC
TRANSMITTER
TRANSMIT HOLDING REGISTER
TRANSMIT SHIFT REGISTER
TxD
XTAL1/BRCLK
XTAL2
RECEIVER
RECEIVE HOLDING REGISTER
RECEIVE SHIFT REGISTER
RxD
INTERRUPTS
TIMING
CONTROLS
BD00120S

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
D0 – D7	16 – 19, 23 – 26	18 – 21, 25 – 28	I/O	**Data Bus:** 8-bit 3-State bidirectional data bus. All data, command and status transfers are made using this bus. D0 is the least significant bit; D7 is the most significant bit.
A0 – A2	31 – 33	33, 35, 36	I	**Address Lines:** Used to select internal PKCC registers or commands.
$\overline{\text{RD}}$	29	31	I	**Read Strobe:** When low, gates the selected PKCC register onto the data bus if $\overline{\text{CE}}$ is also low.
$\overline{\text{WR}}$	28	30	I	**Write Strobe:** When low, gates the contents of the data bus into the selected PKCC register if $\overline{\text{CE}}$ is also low.
$\overline{\text{CE}}$	30	32	I	**Chip Enable:** When high, places the D0 – D7 output drivers in a 3-State condition. If $\overline{\text{CE}}$ is low, data transfers are enabled in conjunction with the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ inputs.
$\overline{\text{INTR}}$	22	24	O	**Interrupt Request:** Several conditions may be programmed to request an interrupt to the CPU. It is an active low open-drain output. This pin will be inactive after power on reset or a master reset command.
$\overline{\text{INTA}}$	27	29	I	**Interrupt Acknowledge:** Used to indicate that an interrupt request has been accepted by the CPU. When $\overline{\text{INTA}}$ goes low, the PKCC outputs an 8-bit address vector on D0 – D7 corresponding to the highest priority interrupt currently active.
$\overline{\text{XINTR}}$	21	23	I	**External Interrupt:** An active low external interrupt input to the PKCC interrupt priority resolver.
TxC	34	37	I/O	**Transmitter Clock:** The function of this pin depends on bit 7 of the baud rate control register (BRR7). If external transmitter clock is selected (BRR7 = 0), it is an input for the transmitter clock. If internal transmitter clock is selected (BRR7 = 1), this pin is an output which is a multiple of the actual baud rate (1×, 16×) as selected by BRR5. The data is transmitted on the falling edge of TxC. It is an input after power on and after master reset or communications reset commands.
RxC	35	38	I/O	**Receiver Clock:** The function of this pin depends on BRR6. If external receiver clock is selected (BRR6 = 0), it is an input for the receiver clock. If internal receiver clock is selected (BRR6 = 1), this pin is an output which is a multiple of the actual baud rate (1×, 16×) as selected by BRR4. The received data is sampled on the rising edge of RxC. It is an input after power on and after master reset or communications reset commands.
TxD	38	41	O	**Transmitter Data:** This output is the transmitted serial data; the least significant bit is transmitted first. This pin is high after power on reset or a reset command that affects the transmitter.
RxD	39	42	I	**Receiver Data:** This input is the serial data input to the receiver. The least significant bit is received first.
XTAL1, XTAL2/BRCLK	36, 37	39, 40	I	**Connections for Crystal:** Provides an on-chip clock generator for the internal baud rate generator and the keyboard interface logic. If an external clock is provided, use XTAL2 as the clock input. See Figures 20 and 21. All timing parameters such as keyboard scan time, tone frequency, and baud rate assume a clock input at the specified BRG input frequency. If this frequency is different the timing parameters will vary proportionately.
KR0 – KR2	10 – 8	11 – 9	O	**Keyboard Row Scan:** Decoded externally; selects one of eight rows.
KC0 – KC3	7 – 4	8 – 5	O	**Keyboard Column Scan:** Decoded externally; selects one of 16 columns.
KRET	15	17	I	**Key Return:** An active high-level indicates that the key being scanned is closed.
$\overline{\text{SHIFT}}$	12	14	I	**SHIFT Key:** Active low input from the SHIFT key. The combination of $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$ inputs select one of four possible codes from the internal key encoding ROM.
$\overline{\text{CONTROL}}$	13	15	I	**CONTROL Key:** Active low input from the CONTROL key. The combination of $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$ inputs select one of four possible codes from the internal key encoding ROM.
$\overline{\text{REPEAT}}$	11	13	I	**REPEAT Key:** Active low input from the $\overline{\text{REPEAT}}$ key. Causes the key depression currently active to be repeated at a rate of approximately 15 times per second.
KCLK	3	4	O	**Keyboard Clock:** High frequency (approximately 400kHz) output used to scan capacitive keyboards.
KDRES	2	3	O	**Key Detect Reset:** Resets the analog detector before scanning a key. Used for capacitive keyboards.
$\overline{\text{HYS}}$	1	2	O	**Hysteresis Output:** Sent to the analog detector for capacitive keyboard applications. A low indicates the key currently being scanned has been recognized on previous scan cycles.
TONE	14	16	O	**Square Wave Output:** Used for tone generation.
V_{CC}	40	44	I	**Power Supply:** +5V.
V_{SS}	20	22	I	Ground.

Programmable Keyboard and Communication Controller (PKCC)

SCN2671

The PKCC has an interrupt mask register to selectively enable certain keyboard and communications status bits to generate interrupts. Priority encoded interrupt vectoring is available. Upon receipt of an interrupt acknowledge, an interrupt vector will be output on D0 – D7 reflecting the source of the interrupt. The interrupt source can also be read from an interrupt status register.

Table 1. Register Addressing

$\overline{\text{CE}}$	A2	A1	A0	$\overline{\text{RD}}/\overline{\text{WR}}$	FUNCTION
1	X	X	X	X	3-State data bus
0	0	0	0	$\overline{\text{WR}}$	Reset command (see Table 6)
0	0	0	0	$\overline{\text{RD}}$	Read interrupt status register (ISR)
0	0	0	1	$\overline{\text{RD}}$, $\overline{\text{WR}}$	Read/write communications mode register (CMR)
0	0	1	0	$\overline{\text{WR}}$	Write transmit holding register (TxHR)
0	0	1	0	$\overline{\text{RD}}$	Read receiver holding register (RxHR)
0	0	1	1	$\overline{\text{WR}}$	Write baud rate mode register (BRR)
0	0	1	1	$\overline{\text{RD}}$	Read communications status register (CSR)
0	1	0	0	$\overline{\text{RD}}$, $\overline{\text{WR}}$	Read/write interrupt mask register (IMR)
0	1	0	1	$\overline{\text{RD}}$, $\overline{\text{WR}}$	Read/write keyboard mode register (KMR)
0	1	1	0	$\overline{\text{RD}}$	Read keyboard holding register (KHR)
0	1	1	1	$\overline{\text{RD}}$	Read keyboard status register (KSR)
0	1	1	1	$\overline{\text{WR}}$	Miscellaneous commands (see Description)

NOTE:
X = don't care.

FUNCTIONAL DESCRIPTION

The PKCC consists of six major sections (see block diagram). These are the transmitter, receiver, timing, operation control, keyboard encoder, and a priority encoded interrupt control unit. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a bidirectional data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers KMR and CMR, the command decoder, and status registers KSR and CSR. Details of operating modes and status information are presented in the Operation section of this data sheet. The register addressing is specified in Table 1.

Timing

The PKCC contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 baud rates, any of which can be selected for full duplex operation. The external clock to the baud rate generator can be applied directly to the XTAL2 input (see Figure 21) or can be generated internally by connecting a crystal across the XTAL1, XTAL2 input pins. The clock input is also utilized by the keyboard encoder section. Thus, a clock must be provided even if external transmitter and receiver clocks are used.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for break conditions, framing and parity errors, and loads an "assembled" character in the receive holding register for access by the CPU.

Transmitter

The transmitter accepts parallel data loaded by the CPU into the transmit holding register and converts it to a serial bit stream framed by the start bit, calculated parity bit (if specified), and stop bit(s). The composite serial stream of data is transmitted on the TxD output pin.

Keyboard Encoder

The keyboard encoder provides encoded scanning signals for a matrix keyboard. Key depressions are detected on the KRET input. The debounced and verified key codes (or matrix addresses) are loaded into the key holding register for access by the CPU. Figures 1 and 2 illustrate the PKCC interface to contact and capacitive keyboards, respectively.

Interrupt Control

The interrupt controller unit contains a software-programmable interrupt mask register which selectively enables status conditions from the keyboard encoder and communication controller to generate interrupts. The interrupts are priority encoded and individually generate an eight bit vector which is output on the data bus in response to a CPU interrupt acknowledge on the $\overline{\text{INTA}}$ input pin.

OPERATION

Keyboard Encoder

The keyboard is continuously scanned by KC0 – KC3 and KR0 – KR2 which are decoded externally to handle 128 possible keys (see Figures 1 and 2). KC0 – KC3 select one of 16 columns and KR0 – KR2 multiplex the eight row return lines into the KRET pin. Debouncing is accomplished by remembering a 1 state at the KRET pin when a key is being addressed and verifying it one scan later. Once the key is verified, a key code is loaded into the keyboard data register (KDR). If the keyboard holding register (KHR) is empty, the contents of the KDR will be transferred to the KHR immediately; if the KHR is full (i.e., the CPU has not read the previous key code), the transfer will be held off until the KHR is read. The data transfer to the KHR causes keyboard data ready (KRDY) to be set in the keyboard status register.

For capacitive keyboards, the high frequency output KCLK can be used to gate the column scan to the keyboard (see Figure 2). The key detector reset (KDRES) output resets the analog detector prior to scanning each key location. The output from the analog multiplexer is sensed and then latched in the analog detector. The $\overline{\text{HYS}}$ output controls the sense level. A 0 will lower the sense level causing hysteresis, and a 1 will raise the sense level with no hysteresis.

The $\overline{\text{REPEAT}}$ input enables the keyboard logic to recognize any key repeatedly, 15 times per second. Additionally, certain keys can be programmed to repeat automatically if depressed for more than one-half second.

A square wave is output on the TONE pin when the CPU issues a ring tone command to the PKCC.

Keyboard Mode Register

Operating modes are selected by programming the keyboard mode register (KMR), see Figure 3. Bit KMR7 is used for testing the device. For normal operation, this bit should always be written to a 0. Bits KMR6 – KMR5 select the rollover modes for keyboard processing:

N-key Rollover: In this mode, the code corresponding to each key depression is loaded into the KDR as soon as that key is debounced, independent of the release of other keys. Two or more closures occurring within one scan cycle are considered to be simultaneous, which will set keyboard error in the keyboard status register (KSR1). As soon as the keyboard holding register is empty, the code in the KDR is transferred to the KHR and the KRDY status bit is set (KSR0).

N-Key Rollover With Latched Keys: This mode is the same as regular N-key rollover, except that the keys which are assigned to

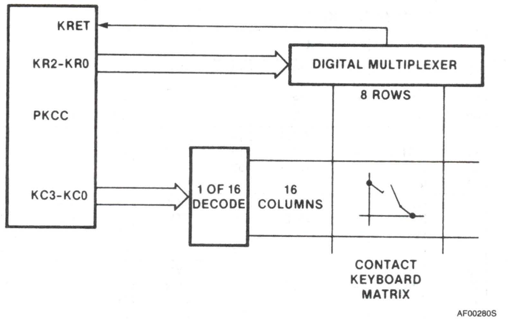

KRET
KR2-KR0
DIGITAL MULTIPLEXER
8 ROWS
PKCC
KC3-KC0
1 OF 16 DECODE
16 COLUMNS
CONTACT KEYBOARD MATRIX
AF00280S

Figure 1. Contact Keyboard Interface

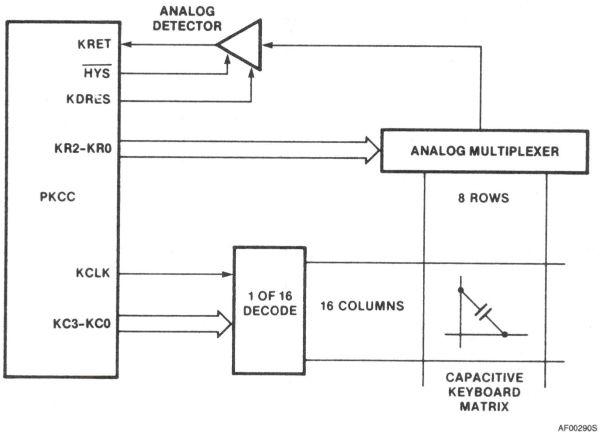

ANALOG DETECTOR
KRET
$\overline{\text{HYS}}$
KDRES
KR2-KR0
ANALOG MULTIPLEXER
PKCC
8 ROWS
KCLK
KC3-KC0
1 OF 16 DECODE
16 COLUMNS
CAPACITIVE KEYBOARD MATRIX
AF00290S

Figure 2. Capacitive Keyboard Interface

row 0 of the keyboard matrix (KR2 – KR0 = 000) produce a code both when depressed and when released. The codes are independent of the states of the inputs at $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$. If one or more of the latched keys are depressed when the keyboard is enabled (after a keyboard reset), the corresponding codes will be sent out as the keys are scanned and debounced. Note that simultaneous latched keys will not set KERR (KSR1) and that latched keys will not be auto-repeat and will not be affected by the $\overline{\text{REPEAT}}$ input.

Two-Key Rollover: The first key code is loaded into the KDR immediately and the second code is loaded only after the first key is released. Simultaneous keys will set KERR (KSR1), if three or more keys remain closed at any given time, the KERR bit will also be set. All keys must then be released before the next KRET will be processed.

Two-Key Inhibit: All keys must be released between keystrokes; otherwise, KERR (KSR1) will be set.

Bit KMR4 specifies the key encoding mode. Each key is assigned four 8-bit codes, corresponding to the states of the $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$ inputs. If the encoded mode is programmed, the row/column address of the detected key is used to load one of the four key codes into the KDR. See Table 2 for key code assignments. If the non-encoded mode is programmed, the row/column address is loaded directly into the KDR with the following format:

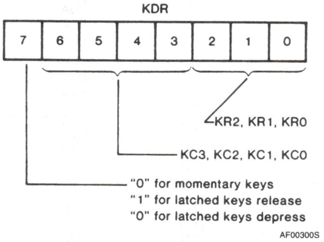
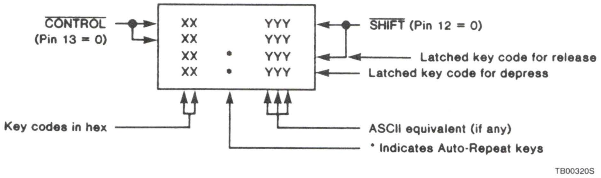

KDR
7
6
5
4
3
2
1
0
KR2, KR1, KR0
KC3, KC2, KC1, KC0
"0" for momentary keys
"1" for latched keys release
"0" for latched keys depress
AF00300S

Table 2. Standard Key Codes (HEX)

COLUMN (KC3 – KC0)	ROW (KR2 – KR0)							
	0	1	2	3	4	5	6	7
0	E0	C0	1B ESC	09 HT	1F US	1A SUB	30 0	2B +
	F0	D0	1B ESC	09 HT	1F US	1A SUB	30 0	3B ;
	E0	C0	1B ESC	09 * HT	1F US	5A Z	30 0	2B +
	F0	D0	1B ESC	09 * HT	1F US	7A z	30 0	3B ;
1	E1	C1	21 !	11 DC1	01 SOH	18 CAN	3D =	2A *
	F1	D1	31 1	11 DC1	01 SOH	18 CAN	2D –	3A :
	E1	C1	21 !	51 Q	41 A	58 X	3D * =	2A *
	F1	D1	31 1	71 q	61 a	78 x	2D * –	3A :
2	E2	C2	22 "	17 ETB	13 DC3	03 ETX	1E RS	1F US
	F2	D2	32 2	17 ETB	13 DC3	03 ETX	1E RS	1F US
	E2	C2	22 "	57 W	53 S	43 C	7E ~	7F * DEL
	F2	D2	32 2	77 w	73 s	63 c	5E ↑	5F * —
3	E3	C3	23 #	05 ENQ	04 EOT	16 SYN	1C FS	1B ESC
	F3	D3	33 3	05 ENQ	04 EOT	16 SYN	1C FS	1B ESC
	E3	C3	23 #	45 E	44 D	56 V	7C ¦	7B {
	F3	D3	33 3	65 e	64 d	76 v	5C \	5B [
4	E4	C4	24 $	12 DC2	06 ACK	02 STX	08 BS	1D GS
	F4	D4	34 4	12 DC2	06 ACK	02 STX	08 BS	1D GS
	E4	C4	24 $	52 R	46 F	42 B	08 * BS	7D }
	F4	D4	34 4	72 r	66 f	62 b	08 * BS	5D]
5	E5	C5	25 %	14 DC4	07 BEL	0E SO	10 DLE	08 BS
	F5	D5	35 5	14 DC4	07 BEL	0E SO	10 DLE	08 BS
	E5	C5	25 %	54 T	47 G	4E N	50 P	08 BS
	F5	D5	35 5	74 t	67 g	6E n	70 p	08 * BS
6	E6	C6	26 &	19 EM	08 BS	0D CR	00 NUL	09 HT
	F6	D6	36 6	19 EM	08 BS	0D CR	00 NUL	09 HT
	E6	C6	26 &	59 Y	48 H	4D M	60 `	09 * HT
	F6	D6	36 6	79 y	68 h	6D m	40 @	09 * HT
7	E7	C7	27 ’	15 NAK	0A LF	3C <	7F DEL	20 SP
	F7	D7	37 7	15 NAK	0A LF	2C ,	7F DEL	20 SP
	E7	C7	27 ’	55 U	4A J	3C <	7F DEL	20 * SP
	F7	D7	37 7	75 u	6A j	2C ,	7F DEL	20 * SP
8	E8	C8	28 (	09 HT	0B VT	3E >	0A LF	0B VT
	F8	D8	38 8	09 HT	0B VT	2E .	0A LF	0B VT
	E8	C8	28 (	49 I	4B K	3E >	0A LF	0B * VT
	F8	D8	38 8	69 i	6B k	2E .	0A LF	0B * VT
9	E9	C9	29)	0F SI	0C FF	3F ?	0D CR	0A LF
	F9	D9	39 9	0F SI	0C FF	2F /	0D CR	0A LF
	E9	C9	29)	4F O	4C L	3F ?	0D CR	0A * LF
	F9	D9	39 9	6F o	6C l	2F /	0D CR	0A * LF
A	EA	CA	37 7	34 4	31 1	30 0	A0	A6
	FA	DA	37 7	34 4	31 1	30 0	B0	B6
	EA	CA	37 7	34 4	31 1	30 0	A0	A6
	FA	DA	37 7	34 4	31 1	30 0	B0	B6
B	EB	CB	38 8	35 5	32 2	2E .	A1	A7
	FB	DB	38 8	35 5	32 2	2E .	B1	B7
	EB	CB	38 8	35 5	32 2	2E .	A1	A7
	FB	DB	38 8	35 5	32 2	2E .	B1	B7
C	EC	CC	39 9	36 6	33 3	BF	A2	A8
	FC	DC	39 9	36 6	33 3	AF	B2	B8
	EC	CC	39 9	36 6	33 3	9F	A2	A8
	FC	DC	39 9	36 6	33 3	8F	B2	B8
D	ED	CD	90	93	82	95	A3	A9
	FD	DD	90	93	82	95	B3	B9
	ED	CD	90	93	82 *	95	A3 *	A9 *
	FD	DD	90	93	82 *	95	B3 *	B9 *
E	EE	CE	91	80	84	81	A4	AA
	FE	DE	91	80	84	81	B4	BA
	EE	CE	91	80 *	84	81 *	A4 *	AA *
	FE	DE	91	80 *	84	81 *	B4 *	BA *
F	EF	CF	92	94	83	96	A5	AB
	FF	DF	92	94	83	96	B5	BB
	EF	CF	92	94	83 *	96	A5 *	AB *
	FF	DF	92	94	83 *	96	B5 *	BB *

This row contains the latched keys when that mode is selected (KMR6, KMR5 = 00)


$\overline{\text{CONTROL}}$ (Pin 13 = 0)
XX
XX
XX •
XX •
YYY
YYY
YYY
YYY
$\overline{\text{SHIFT}}$ (Pin 12 = 0)
Latched key code for release
Latched key code for depress
Key codes in hex
ASCII equivalent (if any)
* Indicates Auto-Repeat keys
TB00320S

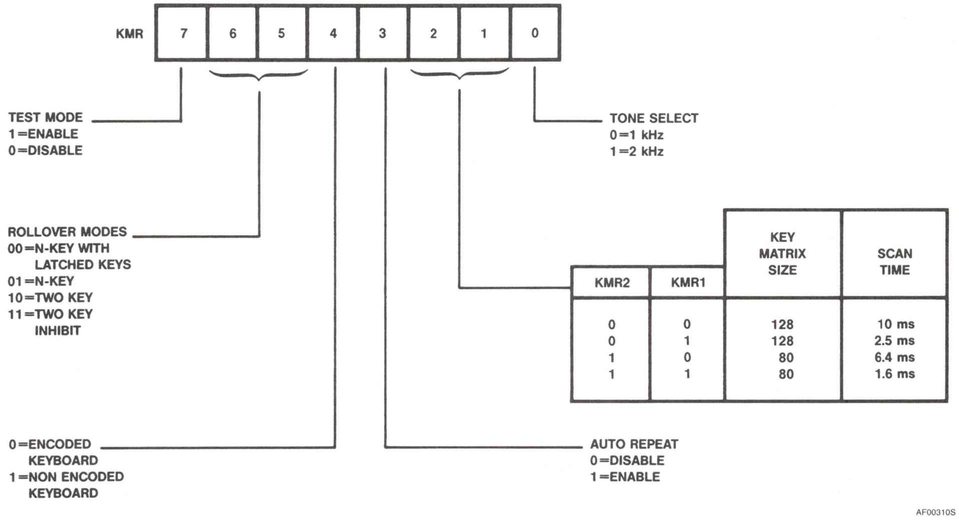

KMR
7
6
5
4
3
2
1
0
TEST MODE
1=ENABLE
0=DISABLE
TONE SELECT
0=1 kHz
1=2 kHz
ROLLOVER MODES
00=N-KEY WITH LATCHED KEYS
01=N-KEY
10=TWO KEY
11=TWO KEY INHIBIT
KMR2
KMR1
KEY MATRIX SIZE
SCAN TIME
0 0 128 10 ms
0 1 128 2.5 ms
1 0 80 6.4 ms
1 1 80 1.6 ms
0=ENCODED KEYBOARD
1=NON ENCODED KEYBOARD
AUTO REPEAT
0=DISABLE
1=ENABLE
AF00310S

Figure 3. Keyboard Mode Register Format

Bit KMR3 enables the auto-repeat mode. In this mode, if a key that is programmed for auto-repeat is depressed for longer than one-half second, the key code will be loaded into the KDR approximately 15 times per second until that key is released. Only the non-control key codes will auto-repeat, i.e. $\overline{\text{CONTROL}}$ = 1. Table 2 specifies the auto-repeat keys.

KMR2 and KMR1 select the key matrix size and debounce time (scan rate). The keyboard row outputs (KR2, KR1, KR0) always scan from 0 to 7. The column outputs (KC3, KC2, KC1, KC0) scan from 0 to 15 for a 128 key matrix and from 0 to 9 for an 80 key matrix.

KMR0 selects between a 1kHz and 2kHz frequency to be output on the TONE pin in response to a ring tone command.

Keyboard Status Register

The keyboard status register (KSR) provides operational feedback to the CPU. Its format is illustrated in Figure 4.

KSR7, 6 and 4 reflect the state of the inputs at the corresponding pins. $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are latched at the time the key is accepted. As the verified codes are loaded into the KDR, the corresponding states of $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are loaded into the KSR. $\overline{\text{REPEAT}}$ is updated on every matrix sample. The status bits are the complements of the input levels.

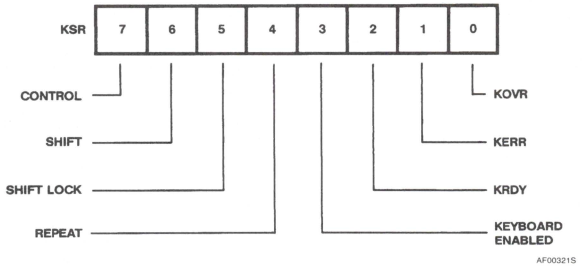

KSR
7
6
5
4
3
2
1
0
CONTROL
SHIFT
SHIFT LOCK
REPEAT
KOVR
KERR
KRDY
KEYBOARD ENABLED
AF00321S

Figure 4. Keyboard Status Register Format

KSR5 reflects the state of the internal shift lock flag which is controlled by the set/reset shift lock commands.

KSR3 indicates that the keyboard controller is enabled. It is controlled by the set/clear keyboard enable command.

Keyboard overrun (KSR2) is set when both the KHR and KDR are full and a third key is validated. The original content of the KHR is preserved and the content of the KDR is overwritten with the new key code. This bit can be specified (by IMR1) to generate an interrupt and is cleared by the reset command with D2 = 1.

Keyboard error (KSR1) is set when the operator depresses more keys than are allowed in the selected rollover mode, or when keys are depressed simultaneously (within one scan cycle). This bit can be specified (by IMR3) to generate an interrupt and is cleared by the reset command with D1 = 1.

Keyboard data ready (KSR0) is set when the key code or address is transferred from the KDR to the KHR. This bit can be specified (by IMR2) to generate an interrupt. It is cleared when the CPU reads the KHR.

Communications Controller

The communications controller section of the PKCC comprises a full duplex asynchronous receiver/transmitter (UART) with a baud rate generator. Registers associated with these elements are the communications mode register (CMR), the baud rate control register (BRR), and the communications status register (CSR).

Receiver

The receiver accepts serial data on the RxD pin, converts the serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition, and presents the assembled character to the CPU. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled again after a delay of one half of the bit time. If RxD is then high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RxHR) and the RxRDY bit in the CSR is set to a 1. If the character length is less than eight bits, the most significant unused bits in the RxHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (i.e. framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the space is interpreted as a start bit.

The parity error, framing error and overrun error (if any) are strobed into the CSR at the received character boundary. If a break condition is detected (RxD is low for the entire character including the stop bit) only one character consisting of all zeros will be transferred to the RxHR and the received break bit in the CSR is set to 1 (RxRDY is not set when a break is received). The RxD input must return to a high condition for one bit time before a search for the next start bit begins.

Transmitter

The transmitter accepts parallel data from the CPU and converts it to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the transmit holding register (TxHR), the TxD output remains high and the TxEMT bit in the CSR will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the TxHR. The transmitter can be forced to send a continuous low condition by a transmit break command.

If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out.

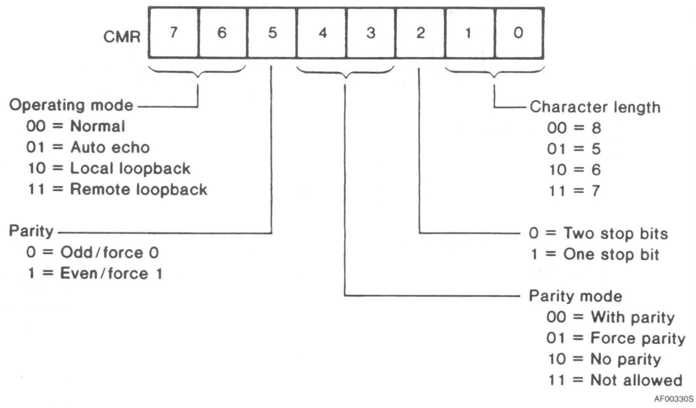


Figure 5. Communications Mode Register Format

Communication Mode Register

Figure 5 illustrates the bit format of the CMR, which controls the operational mode of the communications controller and the character parameters.

Bits CMR1 – CMR0 select a character length of 5, 6, 7, or 8 bits. The character length does not include the parity, start, or stop bits.

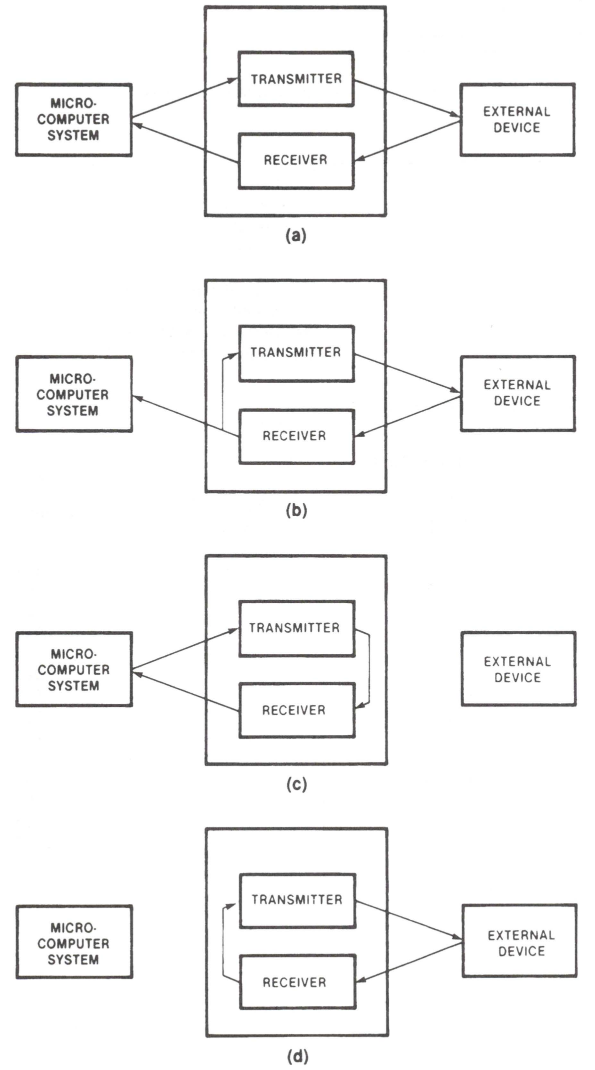
CMR2 selects the transmitted character framing as one or two stop bits. The receiver always checks for one stop bit.

The parity format is selected by bits CMR4 and CMR3. If parity or force parity is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. CMR5 selects odd or even parity and determines the polarity of the parity bit in the force parity mode.

The bits in the mode register affecting character assembly and disassembly (CMR5 – CMR0) can be changed dynamically and affect the characters currently being assembled in RxSR and transmitted by TxSR. To affect assembly of a received character, the CMR must be updated within n–1 bit times of the receipt of that character's start bit. To affect a transmitted character, the CMR must be updated within n–1 bit times of transmitting that character's start bit. (n = the smaller of the new and old character lengths).

The UART can operate in one of four modes, as illustrated in Figure 6. The operating modes are selected by bits CMR7 and CMR6, which should only be changed when both the transmitter and receiver are disabled. CMR7-CMR6 = 00 is the normal mode, with the transmitter and receiver operating independently. CMR7-CMR6 = 01 places the UART in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. Status bit TxRDY is not set. TxEMT operates normally.
5. The receiver parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Only the first character of a break condition is echoed; the TxD output will go high until the next received character is assembled.
7. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.



NOTES:
(a) Normal operating mode.
(b) Automatic echo mode.
(c) Local loopback mode.
(d) Remote loopback mode.

Figure 6. Operating Modes of the 2671 UART

Two diagnostic modes can also be configured. In local loopback mode (CMR7 – CMR6 = 10):

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode (CMR7 – CMR6 = 11). In this mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. No data is sent to the local CPU, but the error status conditions (parity and framing) are set if required.
4. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.

Baud Rate Control Register

The baud rate control register (BRR) controls the frequency generated by the baud rate generator (BRG) and the clock source used by the receiver and transmitter. Its format is illustrated in Figure 7.

BRR3 – BRR0 select one of sixteen frequencies to be generated by the BRG. See Table 3.

BRR7 and BRR6 select the source of the transmit and receive clocks. If external clocks are chosen, (BRR7 = 0 or BRR6 = 0), then the clock rate factor is determined by BRR5 and BRR4. The external clock input(s) should be the desired baud rate multiplied by the clock rate factor.

If internal clock(s) are specified, (BRR7 = 1 or BRR6 = 1), the clock is supplied by the internal baud rate generator at the selected baud rate. The clock rate factor for internally generated clocks is always 16. Pins 35 and 34 become outputs for transmit or receive clocks, respectively. See Table 4 for the description and selection of these outputs.

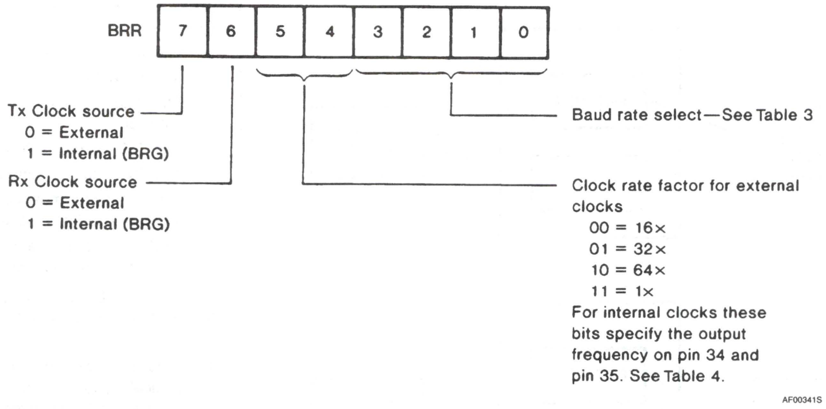

BRR
7
6
5
4
3
2
1
0
Tx Clock source
0 = External
1 = Internal (BRG)
Rx Clock source
0 = External
1 = Internal (BRG)
Baud rate select—See Table 3
Clock rate factor for external clocks
00 = 16×
01 = 32×
10 = 64×
11 = 1×
For internal clocks these bits specify the output frequency on pin 34 and pin 35. See Table 4.
AF00341S

Figure 7. Baud Rate Control Register Format

Table 3. Baud Rate Generator Characteristics (BRCLK = 4.9152MHz)

BRR3 - 0	BAUD RATE	ACTUAL FREQUENCY 16× CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8 kHz	–	6144
0001	110	1.7598	–0.01	2793
0010	134.5	2.152	–	2284
0011	150	2.4	–	2048
0100	200	3.2	–	1536
0101	300	4.8	–	1024
0110	600	9.6	–	512
0111	1050	16.8329	+0.20	292
1000	1200	19.2	–	256
1001	1800	28.7438	–0.20	171
1010	2000	31.9168	–0.26	154
1011	2400	38.4	–	128
1100	4800	76.8	–	64
1101	9600	153.6	–	32
1110	19200	307.2	–	16
1111	38400	614.4	–	8

Table 4. Baud Rate Control Register

BRR7 – BRR4	CLOCK SOURCE		PIN FUNCTIONS		BRR3 – BRR0 BAUD RATE SELECTION
	TxC	RxC	PIN 34	PIN 35	
00**	E	E	TxC	RxC	The baud rates are listed in Table 3.
01**	E	I	TxC	1×	
10**	I	E	16×	RxC	
1100	I	I	1×	1×	
1101	I	I	1×	16×	
1110	I	I	16×	1×	
1111	I	I	16×	16×	

NOTES:

1. ** = Clock rate factor for external clocks: 00 = 16×
   01 = 32×
   10 = 64×
   11 = 1×
2. E = External clock.
3. I = Internal clock (BRG).
4. 1× and 16× are clock outputs at 1 or 16 times the actual baud rate. For receive, the 1× output is the actual data sample clock.
5. BRR7 – BRR6 = 01 or 10 not permitted in automatic echo or remote loopback modes unless BRR5 – BRR4 = 00.

Communications Status Register

Figure 8 illustrates the bit format of the communications status register (CSR), which provides UART status to the CPU.

Receiver ready (CSR0) indicates that a received character is assembled and transferred to the RxHR and is ready to be read by the CPU. This bit can be specified (by IMR0) to generate an interrupt and is reset by reading the RxHR.

Transmitter ready (CSR1) indicates that the TxHR is empty and ready to be loaded with a character. This bit will be cleared when the TxHR is loaded and has not yet transferred the character to the transmit shift register (TxSR). TxRDY is reset when the transmitter is disabled. It will be set when the transmitter is enabled, provided that no data was loaded into the TxHR during the time the transmitter was disabled. This bit can be specified (by IMR7) to generate an interrupt.

Transmitter empty (CSR2) indicates that the transmitter has underrun, i.e., both the TxHR and TxSR are empty. This bit can only be set after transmission of at least one character, and is cleared when the TxHR is loaded by the CPU. TxEMT is reset when the transmitter is disabled. This bit can be specified (by IMR6) to generate an interrupt.

CSR3 will be set when the PKCC receives a command to transmit a break. This bit will be cleared after the break is completed.

Received break (CSR4) indicates that an all zero character of the programmed length has been received without a stop bit. Breaks originating in the middle of a received character can be detected. This bit is cleared when RxD returns to a high state for at least one bit time.

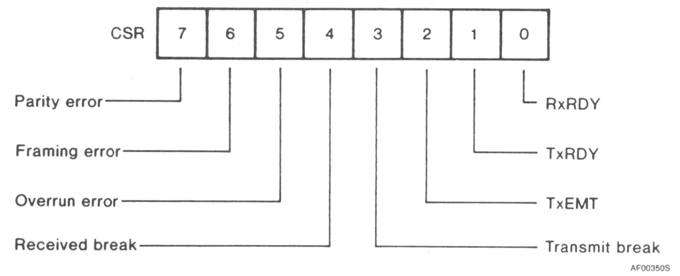


Figure 8. Communications Status Register Format

Receiver overrun (CSR5) indicates that the previous character in the RxHR has not been read by the CPU and that a new character has been loaded into the RxHR. This bit is cleared by a reset command with D3 = 1.

Framing error (CSR6) indicates that the stop bit has not been detected. The stop bit check is made in the middle of the first stop bit position. This bit is cleared by a reset command with D3 = 1.

Parity error (CSR7) indicates that a character was received with incorrect parity when 'with parity' or 'force parity' is enabled. This bit is cleared by a reset command with D3 = 1.

Interrupt Controller

The SCN2671 contains a maskable interrupt status register (ISR) which can be enabled to generate an active low interrupt request on the $\overline{\text{INTR}}$ output. The eight interrupt conditions in the ISR are individually enabled by writing a 1 into the corresponding bit of the interrupt mask register (IMR).

Each of the interrupt conditions is assigned a priority and a vector. When an enabled ISR bit is set, the SCN2671 asserts the $\overline{\text{INTR}}$ output. If the CPU activates the $\overline{\text{INTA}}$ input, the SCN2671 responds by placing the corresponding 8-bit vector on the data bus (D7 – D0). If multiple interrupts are pending, the vector corresponds to the condition with the highest priority. The interrupt will persist until all pending interrupt conditions are cleared.

The ISR can also be polled by reading at address A2 – A0 = 000. All pending interrupt conditions which are enabled by the IMR will be read independent of priority.

The bit assignments of the ISR and IMR and corresponding vectors and priorities are listed in Table 5.

Table 5. Interrupt Mask Register (IMR) and Interrupt Status Register (ISR)

BIT IN IMR/ISR	INTERRUPT CONDITION	PRIORITY	VECTOR ON D7 – D0		CONDITION RESET BY:
			BINARY	HEX	
IMR0/ISR0	RxRDY	1	11001111	CF	Read RxHR
IMR1/ISR1	KOVR	2	11010111	D7	Reset CMD (D2 = 1)
IMR2/ISR2	KRDY	3	11011111	DF	Read KHR
IMR3/ISR3	KERR	4	11100111	E7	Reset CMD (D1 = 1)
IMR4/ISR4	XINT[1]	5	11101111	EF	External
IMR5/ISR5	△BREAK[2]	6	11110111	F7	Reset CMD (D4 = 1)
IMR6/ISR6	TxEMT	7	11000111	C7	Load TxHR
IMR7/ISR7	TxRDY	8	11000111	C7	Load TxHR

NOTES:
1. XINT is an input from an external interrupt source, active low (pin 21).
2. △BREAK refers to the change of a received break condition.

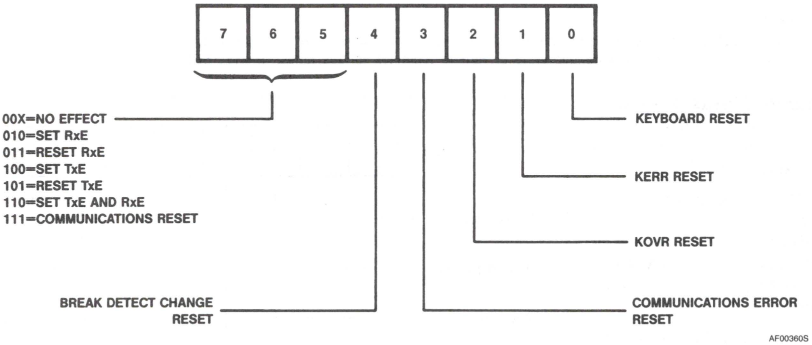


Figure 9. Reset Command Format

COMMANDS

In addition to the control exercised by programming of the PKCC control registers, several functions can be performed by executing command operations. There are two classes of commands which are initiated by writing to the SCN2671 at address A2 – A0 = 000 (reset command) and address A2 – A0 = 111 (miscellaneous commands). Individual commands are specified by the bit pattern on the data bus (D7 – D0).

Reset Commands

The reset command bit format is illustrated in Figure 9 and the detail command descriptions are given in Table 6.

A reset command with D7 – D0 = 111XXXX1 is a master reset for the SCN2671. This command must be given following a power on condition to release the internal power on reset latch which deactivates the SCN2671 on power-up.

Miscellaneous Commands

The miscellaneous command format is illustrated in Figure 10.

The transmit break commands force a break (steady low output) on the TxD pin immediately or after the character in the TxSR (if any) is transmitted. A timed break lasts for approximately 200ms, and a character break lasts for one character time including parity and stop bit time. In either case, TxRDY (CSR1) will be set at the beginning of the break which can be extended indefinitely (by 200ms or one character time increments) by reasserting the command in response to TxRDY. Note that these commands reset TxRDY. When a transmit break command is asserted, CSR3 will be set. The bit will be cleared after the break is completed.

The ring tone commands cause the tone generator to output a square wave on the TONE output. The tone durations are specified by the commands:

Ring tone short = 25ms
Ring tone long = 100ms

The tone frequency is either 1kHz or 2kHz, as specified by KMR0.

Table 6. Reset Command Description

COMMAND	RESETS	COMMENTS
Keyboard reset	KMR7 – KMR0 KSR5, KSR3 – KSR0 IMR3 – IMR1	The keyboard controller is reset, ignoring the input at KRET.
KERR reset	KSR1	Keyboard error status bit reset.
KOVR reset	KSR2	Keyboard overrun status bit reset.
Communications error reset	CSR7 – CSR5	Resets the receiver overrun, parity, and framing error status bits.
Break detect change reset	ISR5	Resets the break detect change bit in the interrupt status register.
Set RxE	See note	Enables receiver operation.
Reset RxE	CSR7 – CSR4, CSR0 See note	Disables the receiver.
Set TxE	See note	Enables transmitter operation.
Reset TxE	CSR3 – CSR1 See note	Disables the transmitter. Sets the TxD output to a 1 after transmitting the character in TxSR.
Communications reset	CMR, CSR, BRR, TxE, RxE, IMR7 – IMR5, IMR0	Resets the communication controller. The RxD input is ignored and the TxD output is set to a 1.
Master reset	CMR, CSR, BRR, TxE, RxE, KMR, KSR5, KSR3 – KSR0, IMR7 – IMR0 Releases the internally latched power on reset.	Resets the keyboard and communication controllers. Inputs at KRET and RxD are ignored and the TxD output is set to a 1.

NOTE:
Command does not affect the CMR or the BRR.

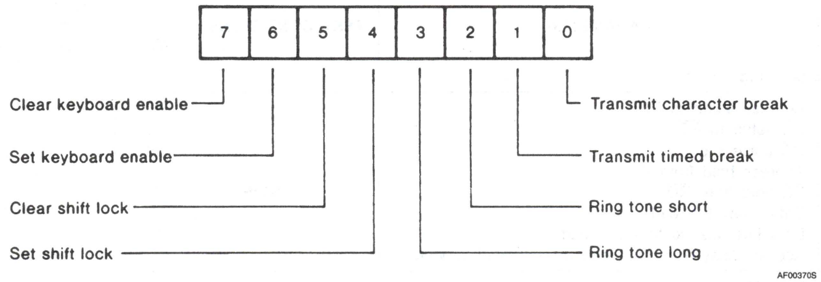


Figure 10. Miscellaneous Commands Format

The set/clear shift lock commands control the state of the internal shift lock flip-flop. When shift lock is set, the keyboard controller encodes all key depressions as if the $\overline{\text{SHIFT}}$ input was asserted. The state of the shift lock flip-flop is reflected in KSR5.

The set keyboard enable command enables the keyboard controller and sets KSR3 in the keyboard status register. The clear keyboard enable command resets KSR3 and disables key processing at the KRET input. The keyboard controller is not reset by this command, and the current state of the keyboard (key depressions and latched key states) is preserved internally. When the keyboard is subsequently enabled, key processing resumes, old and new keys are debounced, and latched keys are encoded if there has been a change in their state.

MASK-PROGRAMMABLE OPTIONS

Characteristics of certain portions of the PKCC are internally programmed by means of a read only memory. The items which can be programmed are:

- Key codes
- Auto-repeat keys
- Scan times, tone frequency, and tone duration
- Baud rates
- Interrupt vectors

Consult your local Signetics representative for costs, minimum quantities, and data submission requirements for customized versions of the PKCC.

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	0 to +70	°C
T_{STG}	Storage temperature range	-65 to +150	°C
	All voltages with respect to ground[3]	-0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage					
	XTAL1, XTAL2/BRCLK		4			V
	All other inputs		2			V
V_{OL}	Output low voltage	$I_{OL} = 1.6mA$			0.4	V
V_{OH}	Output high voltage (except $\overline{INTR}$)	$I_{OH} = -100\mu A$	2.4			V
I_{IL}	Input leakage current (except XTAL1, XTAL2/BRCLK)	$V_{IN} = 0$ to V_{CC}	-10		10	µA
I_{XTLIL}	Input low current	$V_{IN} = 0$				
	XTAL1		-80	-30		µA
	XTAL2/BRCLK[7]		-4	-1.5		mA
I_{XTLIH}	Input high current	$V_{IN} = V_{CC}$				
	XTAL1			30	80	µA
	XTAL2/BRCLK[7]			0.2	1	mA
I_{LL}	Data bus 3-State leakage current	$V_O = 0$ to V_{CC}	-10		10	µA
I_{CC}	Power supply current				150	mA

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Read timing (see Figure 11)						
t_{AS}	Address setup to $\overline{RD}$		50			ns
t_{CS}	$\overline{CE}$ setup to $\overline{RD}$		50			ns
t_{PW}	$\overline{RD}$ pulse width		250			ns
t_{AH}	Address hold from $\overline{RD}$		20			ns
t_{CH}	$\overline{CE}$ hold from $\overline{RD}$		0			ns
t_{DD}	Data delay for read	$C_L = 150pF$			200	ns
t_{DF}	Data bus floating time for read	$C_L = 150pF$	10		100	ns
t_{AD1}	Access delay from any read to next read or write		250			ns
Write timing (see Figure 12)						
t_{AS}	Address setup to $\overline{WR}$		50			ns
t_{CS}	$\overline{CE}$ setup to $\overline{WR}$		50			ns
t_{PW}	$\overline{WR}$ pulse width		250			ns
t_{AH}	Address hold from $\overline{WR}$		20			ns
t_{CH}	$\overline{CE}$ hold from $\overline{WR}$		0			ns
t_{DS}	Data setup		100			ns
t_{DH}	Data hold		10			ns
t_{AD2}	Access delay from any write to next read or write		250			ns
	Access delay from reset command to next read or write		1			µs

AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Interrupt acknowledge timing (see Figure 13)						
t_{PWI}	$\overline{\text{INTA}}$ pulse width		300			ns
t_{DDI}	Data delay time for interrupt vector	$C_L = 150pF$			250	ns
t_{DFI}	Data bus floating time after $\overline{\text{INTA}}$	$C_L = 150pF$	10		100	ns
t_{ADI}	$\overline{\text{INTA}}$ to $\overline{\text{INTA}}$ access delay		300			ns
$\overline{\text{INTR}}$ reset timing (see Figure 14)						
t_{RI}	$\overline{\text{INTR}}$ delay from:					
	Read RxHR (RxRDY)				400	ns
	Read KHR (KRDY)				400	ns
	Reset commands (KOVR, KERR, BREAK)				450	ns
	Load TxHR (TxEMT, TxRDY)				400	ns
	Mask bit reset				300	ns
Keyboard timing (see Figures 15 and 16)						
f_{KCLK}	KCLK frequency			409		kHz
t_{KBD}	KR_i,KC_i to KRET sample delay:					
	FAST SCAN		12			μs
	SLOW SCAN		55			μs
t_{POS}	Scan time per matrix position:					
	FAST SCAN			20		μs
	SLOW SCAN			80		μs
t_{KRD}	KDRES delay from KCLK	$C_L = 150pF$			400	ns
t_{KRH}	KDRES hold from KLCK	$C_L = 150pF$			400	ns
t_{HYSD}	$\overline{\text{HYS}}$ delay from KCLK	$C_L = 150pF$			600	ns
t_{RCD}	KR_i,KC_i delay from KLCK	$C_L = 150pF$			400	ns
UART timing (see Figures 17, 18, 19)						
t_{RXS}	RxD setup time		200			ns
t_{RxH}	RxD hold time		200			ns
t_{TxD}	TxD delay from falling edge of TxC	$C_L = 150pF$			300	ns
t_{TCS}	Skew between TxD transition and falling edge of TxC output	$C_L = 150pF$		0		ns
t_{BRH}	XTAL1 clock high[8]		70			ns
t_{BRL}	XTAL1 clock low[8]		70			ns
f_{BRG}	BRG input frequency		1	4.9152	5.075	MHz
$f_{R/T}$	TxC or RxC input frequency	Clock rate factor = 16×, 32×, 64×			1.3	MHz
		Clock rate factor = 1×			1	MHz
$t_{R/TH}$	TxC or RxC clock high		350			ns
$t_{R/TL}$	TxC or RxC clock low		350			ns

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating on elevated temperatures, the device must be operated based on +150°C maximum function temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground (V_{SS}). All input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum and time measurements are referenced at input voltages of 0.8V, 2V and at output voltages of 0.8V, 2V as appropriate, unless otherwise specified.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. XTAL2 input currents are measured with XTAL grounded.
8. See Figures 20 and 21 for XTAL1, WTAL2 connections for driving XTAL2 with an external clock. Input levels for XTAL1 and XTAL2 are $V_{IL} \leqslant 0.8V$, $V_{IH} \geqslant 4.0V$, and t_{BRL} and t_{BRH} are measured at these levels.

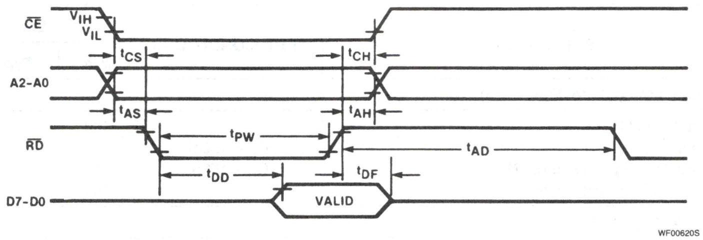

CE
VIH
VIL
tCS
tCH
A2-A0
tAS
tAH
RD
tPW
tAD
tDD
tDF
D7-D0
VALID
WF00620S

Figure 11. Read Timing

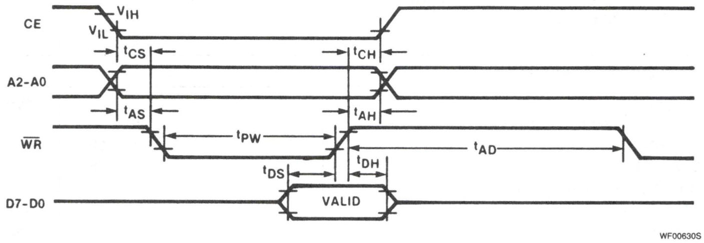

CE
VIH
VIL
tCS
tCH
A2-A0
tAS
tAH
WR
tPW
tAD
tDS
tDH
D7-D0
VALID
WF00630S

Figure 12. Write Timing

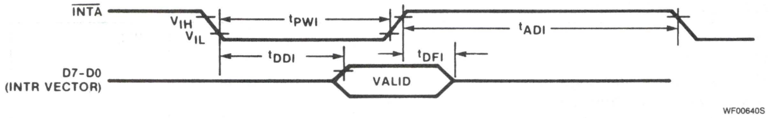

INTA
VIH
VIL
tPWI
tADI
tDDI
tDFI
D7-D0
(INTR VECTOR)
VALID
WF00640S

Figure 13. Interrupt Acknowledge Timing

Programmable Keyboard and Communication Controller (PKCC)

SCN2671

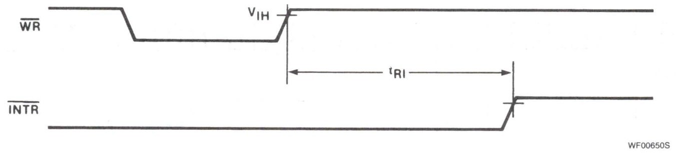
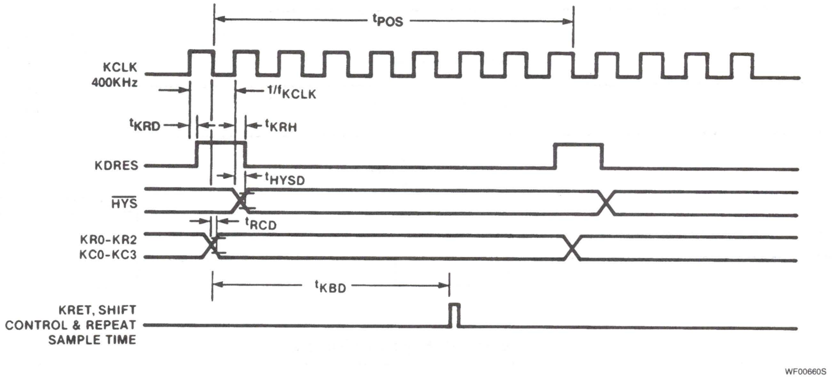

WR
V_IH
t_RI
INTR
WF00650S

Figure 14. Interrupt Reset Timing


t_POS
KCLK
400KHz
1/f_KCLK
t_KRD
t_KRH
KDRES
t_HYSD
HYS
t_RCD
KR0-KR2
KC0-KC3
t_KBD
KRET, SHIFT
CONTROL & REPEAT
SAMPLE TIME
WF00660S

NOTE:
Scan timing shown is for fast scan (KMR1 = 1).
For slow scan (KMR1 = 0) all signals except KCLK run at ¼ the shown rates.

Figure 15. Keyboard Scan Timing

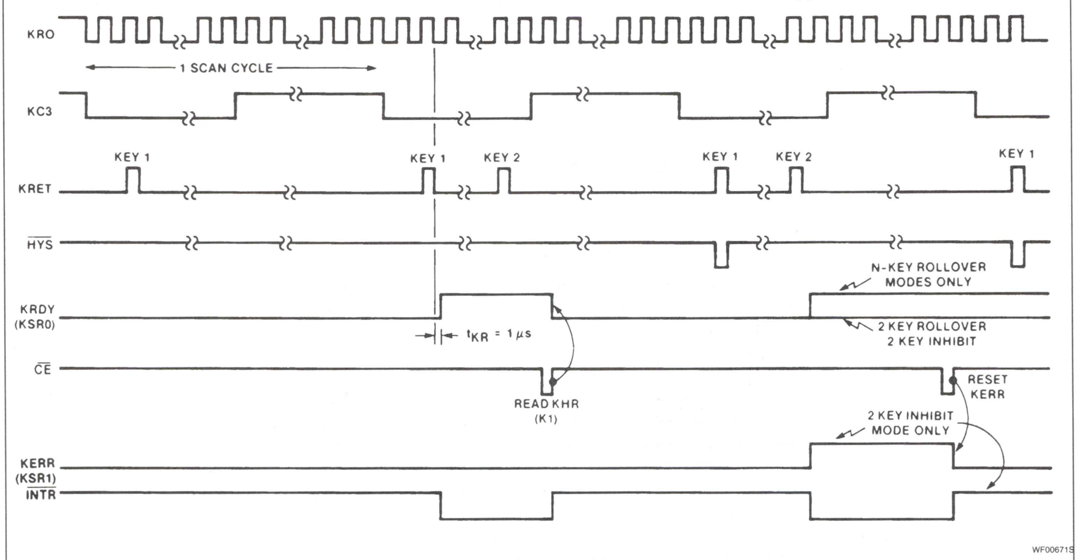
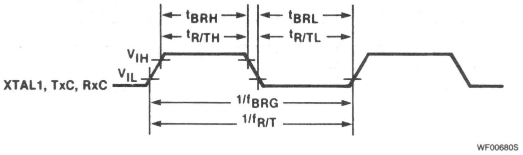
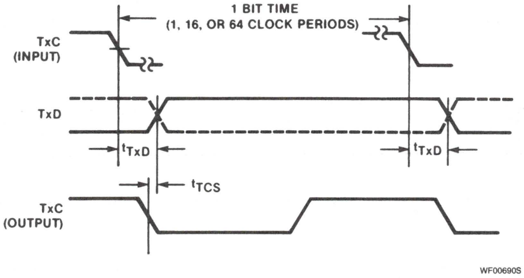

KRO
1 SCAN CYCLE
KC3
KEY 1
KEY 1
KEY 2
KEY 1
KEY 2
KEY 1
KRET
HYS
N-KEY ROLLOVER MODES ONLY
KRDY (KSR0)
t_{KR} = 1 μs
2 KEY ROLLOVER 2 KEY INHIBIT
CE
RESET KERR
READ KHR (K1)
2 KEY INHIBIT MODE ONLY
KERR (KSR1)
INTR
WF00671S

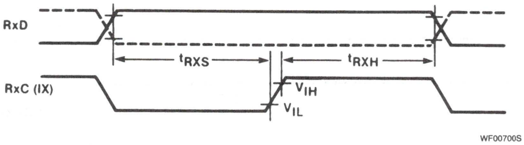
Figure 16. Keyboard Timing


tBRH
tR/TH
tBRL
tR/TL
VIH
VIL
XTAL1, TxC, RxC
1/fBRG
1/fR/T
WF00680S

Clock


1 BIT TIME
(1, 16, OR 64 CLOCK PERIODS)
TxC
(INPUT)
TxD
tTxD
tTxD
tTCS
TxC
(OUTPUT)
WF00690S

Transmit


RxD
tRXS
tRXH
RxC (IX)
VIH
VIL
WF00700S

Receive

Figure 17

Programmable Keyboard and Communication Controller (PKCC)

SCN2671

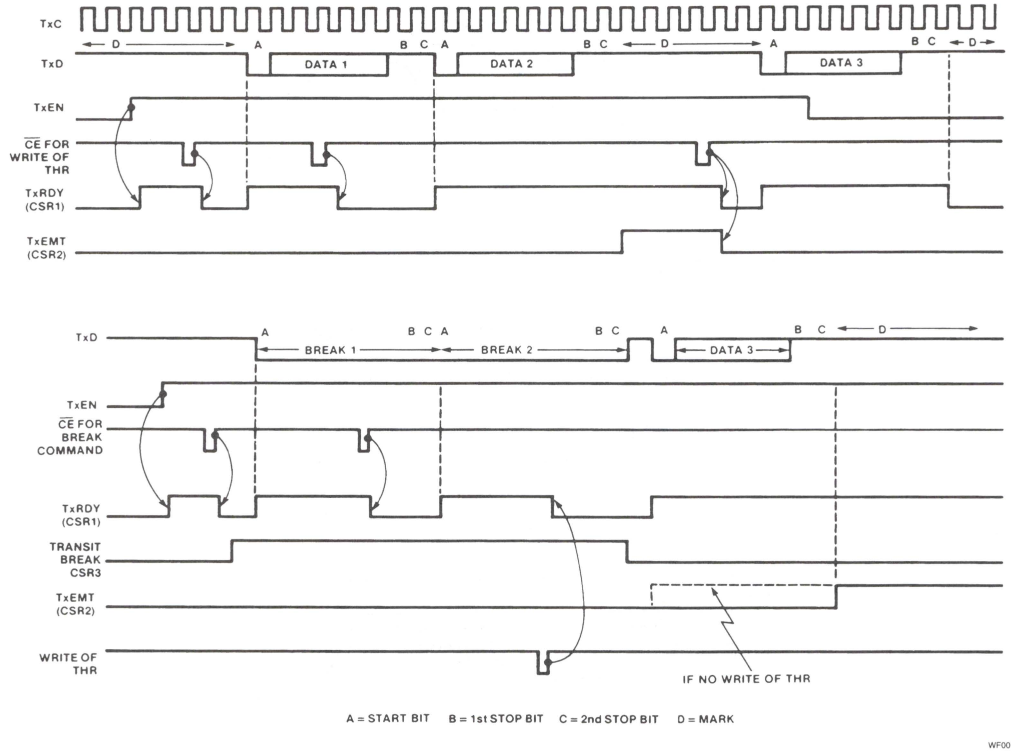

TxC
D
A
B C A
B C
D
A
B C
D
TxD
DATA 1
DATA 2
DATA 3
TxEN
CE FOR
WRITE OF
THR
TxRDY
(CSR1)
TxEMT
(CSR2)
TxD
A
B C A
B C
A
B C
D
BREAK 1
BREAK 2
DATA 3
TxEN
CE FOR
BREAK
COMMAND
TxRDY
(CSR1)
TRANSIT
BREAK
CSR3
TxEMT
(CSR2)
WRITE OF
THR
IF NO WRITE OF THR
A = START BIT
B = 1st STOP BIT
C = 2nd STOP BIT
D = MARK
WF00720S

Figure 18. Transmitter Timing
(5-Bit Characters, No Parity, 2 Stop Bits)

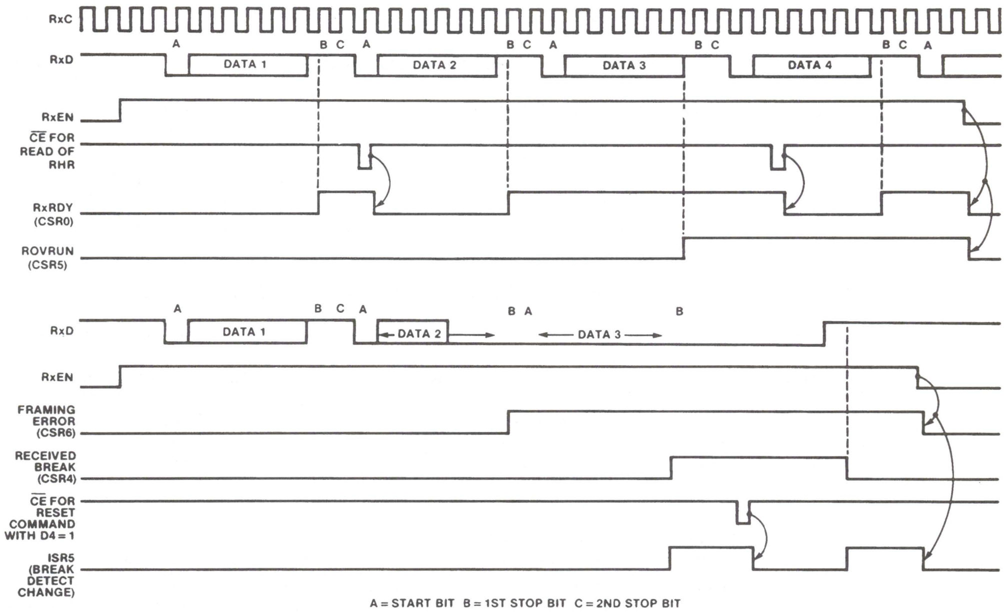

RxC
RxD
A
B C A
B C A
B C A
B C A
DATA 1
DATA 2
DATA 3
DATA 4
RxEN
CE FOR READ OF RHR
RxRDY (CSR0)
ROVRUN (CSR5)
RxD
DATA 1
DATA 2
DATA 3
RxEN
FRAMING ERROR (CSR6)
RECEIVED BREAK (CSR4)
CE FOR RESET COMMAND WITH D4 = 1
ISR5 (BREAK DETECT CHANGE)
A = START BIT B = 1ST STOP BIT C = 2ND STOP BIT
WF00710S

Figure 19. Receiver Timing (5-Bit Characters, No Parity, 2 Stop Bits)

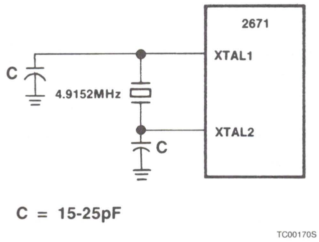

2671
XTAL1
XTAL2
C
C
4.9152MHz
C = 15-25pF
TC00170S

Figure 20. Crystal Connections BRG Clock

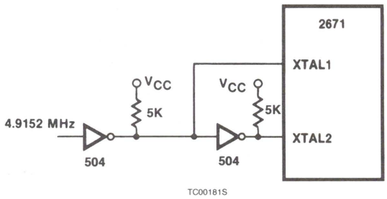

2671
XTAL1
XTAL2
VCC
VCC
5K
5K
4.9152 MHz
504
504
TC00181S

Figure 21. Connection for External BRG Clock Source

Table 7. Register Format Summary

	7	6	5	4	3	2	1	0
	Test Mode	Rollover modes		Keyboard	Auto repeat			Tone select
KMR	1 = Enable 0 = Disable	00 = N-key with latched keys 01 = N-key 10 = Two keys 11 = Two key inhibit		0 = Encoded 1 = Nonencoded	0 = Disable 1 = Enable			0 = 1kHz 1 = 2kHz
KSR	CONTROL	SHIFT	SHIFT LOCK	REPEAT	Keyboard Enabled	KOVR	KERR	KRDY
	Operating Mode		Parity	Parity Mode		Stop Bits	Character Length	
CMR	00 = Normal 01 = Auto echo 10 = Local loopback 11 = Remote loopback		0 = Odd/ force 0 1 = Even/ force 1	00 = With parity 01 = Force parity 10 = No parity 11 = Not allowed		0 = Two 1 = One	00 = 8 01 = 5 10 = 6 11 = 7	
	Tx Clock source	Rx Clock source	Clock rate factor for external clocks		Baud rate select (BRR3 – BRR0 in hex)			
BRR	0 = External 1 = Internal (BRG)	0 = External 1 = Internal (BRG)	00 = 16× 01 = 32× 10 = 64× 11 = 1× For internal clocks these bits specify the output frequency on pins 34 and 35 (Table 4).		0 = 50 1 = 110 2 = 134.5 3 = 150 (BRCLK = 4.9152MHz)	4 = 200 5 = 300 6 = 600 7 = 1050	8 = 1200 9 = 1800 A = 2000 B = 2400	C = 4800 D = 9600 E = 19200 F = 38400
CSR	Parity error	Framing error	Overrun error	Received break	Transmit break	TxEMT	TxRDY	RxRDY
IMR/ISR	TxRDY	TxEMT	BREAK CHANGE	XINT	KERR	KRDY	KOVR	RxRDY
Reset Command Format	00X = No effect 010 = Set RxE 011 = Reset RxE 100 = Set TxE 101 = Reset TxE 110 = Set TxE and RxE 111 = Communications reset			Break detect change reset	Communications error reset	KOVR reset	KERR reset	Keyboard reset
Miscellaneous Commands Format	Clear keyboard enable	Set keyboard enable	Clear shift lock	Set shift lock	Ring tone long	Ring tone short	Transmit timed break	Transmit character break

KMR2	KMR1	Key Matrix Size	Scan Time
0	0	128	10ms
0	1	128	2.5ms
1	0	80	6.4ms
1	1	80	1.6ms

Signetics

SCN2672
Programmable Video Timing Controller (PVTC)

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics SCN2672 Programmable Video Timing Controller (PVTC) is a programmable device designed for use in CRT terminals and display systems that employ raster scan techniques. The PVTC generates the vertical and horizontal timing signals necessary for the display of interlaced or non-interlaced data on a CRT monitor. It provides consecutive addressing to a user-specified display buffer memory domain and controls the CPU-display buffer interface for various buffer configuration modes. A variety of operating modes, display formats, and timing profiles can be implemented by programming the control registers in the PVTC.

A minimum CRT terminal system configuration consists of a PVTC, an SCN2671 Keyboard and Communication Controller (PKCC), an SCN2670 Display Character and Graphics Generator (DCGG), an SCB2673/2677 Video and Attributes Controller (VAC), a single-chip microcomputer such as the 8048, a display buffer RAM, and a small amount of TTL for miscellaneous address decoding, interface, and control. Typically, the package count for a minimum system is between 15 and 20 devices; system complexity can be enhanced by upgrading the microprocessor and expanding via the system address and data buses.

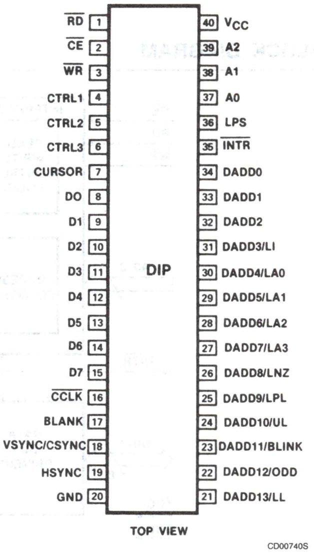
FEATURES

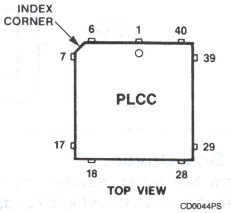
- **4MHz and 2.7MHz character rate versions**
- **Up to 256 characters per row**
- **1 to 16 raster lines per character row**
- **Up to 128 character rows per frame**
- **Programmable horizontal and vertical sync generators**
- **Interlaced or non-interlaced operation**
- **Up to 16k RAM addressing for multiple page operation**
- **Automatic wraparound of RAM**
- **Addressable incrementable and readable cursor**
- **Programmable cursor size, position, and blink**
- **Split screen and horizontal scroll capability**
- **Light pen register**
- **Selectable buffer interface modes**
- **Dynamic RAM refresh**
- **Completely TTL compatible**
- **Single +5V power supply**
- **Power-on reset circuit**

APPLICATIONS

- **CRT terminals**
- **Word processing systems**
- **Small business computers**
- **Home computers**

PIN CONFIGURATIONS


$\overline{\text{RD}}$ 1
$\overline{\text{CE}}$ 2
$\overline{\text{WR}}$ 3
CTRL1 4
CTRL2 5
CTRL3 6
CURSOR 7
D0 8
D1 9
D2 10
D3 11
D4 12
D5 13
D6 14
D7 15
$\overline{\text{CCLK}}$ 16
BLANK 17
VSYNC/CSYNC 18
HSYNC 19
GND 20
40 V_{CC}
39 A2
38 A1
37 A0
36 LPS
35 $\overline{\text{INTR}}$
34 DADD0
33 DADD1
32 DADD2
31 DADD3/LI
30 DADD4/LA0
29 DADD5/LA1
28 DADD6/LA2
27 DADD7/LA3
26 DADD8/LNZ
25 DADD9/LPL
24 DADD10/UL
23 DADD11/BLINK
22 DADD12/ODD
21 DADD13/LL
DIP
TOP VIEW
CD00740S


INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

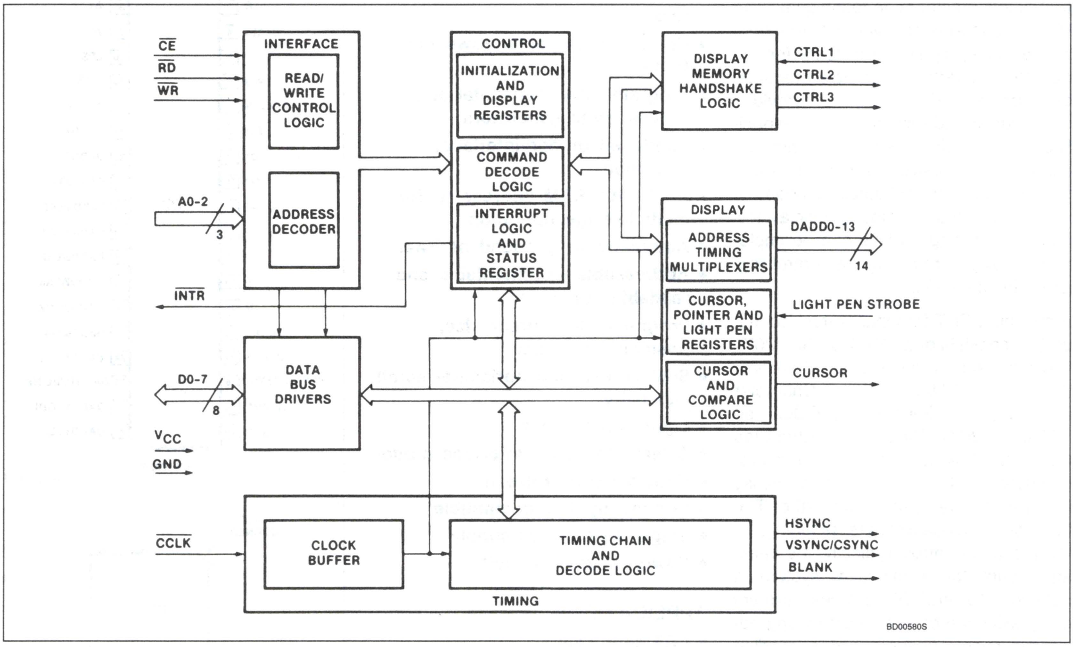
Pin	Function	Pin	Function
1	NC	23	NC
2	$\overline{\text{RD}}$	24	DADD13/LL
3	$\overline{\text{CE}}$	25	DADD12/ODD
4	$\overline{\text{WR}}$	26	DADD11/BLINK
5	CTRL1	27	DADD10/UL
6	CTRL2	28	DADD9/LPL
7	CTRL3	29	DADD8/LNZ
8	CURSOR	30	DADD7/LA3
9	D0	31	DADD6/LA2
10	D1	32	DADD5/LA1
11	D2	33	DADD4/LA0
12	NC	34	NC
13	D3	35	DADD3/LI
14	D4	36	DADD2
15	D5	37	DADD1
16	D6	38	DADD0
17	D7	39	$\overline{\text{INTR}}$
18	$\overline{\text{CCLK}}$	40	LPS
19	BLANK	41	A0
20	VSYNC/CSYNC	42	A1
21	HSYNC	43	A2
22	GND	44	V_{CC}

Programmable Video Timing Controller (PVTC) SCN2672

ORDERING INFORMATION

PACKAGES	$V_{CC} = +5V \pm 5\%$, $T_A = 0°C$ to $+70°C$	
	4MHz	2.7MHz
Ceramic DIP	SCN2672BC4I40	SCN2672BC3I40
Plastic DIP	SCN2672BC4N40	SCN2672BC3N40
Plastic LCC	SCN2672BC4A44	SCN2672BC3A44

BLOCK DIAGRAM


CE
RD
WR
INTERFACE
READ/ WRITE CONTROL LOGIC
A0-2
3
ADDRESS DECODER
INTR
D0-7
8
DATA BUS DRIVERS
VCC
GND
CCLK
CONTROL
INITIALIZATION AND DISPLAY REGISTERS
COMMAND DECODE LOGIC
INTERRUPT LOGIC AND STATUS REGISTER
DISPLAY MEMORY HANDSHAKE LOGIC
CTRL1
CTRL2
CTRL3
DISPLAY
ADDRESS TIMING MULTIPLEXERS
DADD0-13
14
CURSOR, POINTER AND LIGHT PEN REGISTERS
LIGHT PEN STROBE
CURSOR AND COMPARE LOGIC
CURSOR
CLOCK BUFFER
TIMING CHAIN AND DECODE LOGIC
TIMING
HSYNC
VSYNC/CSYNC
BLANK
BD00580S

FUNCTIONAL DESCRIPTION

As shown on the block diagram, the PVTC contains the following major blocks:

- Data bus buffer
- Interface Logic
- Operation Control
- Timing
- Display Control
- Buffer Control

Data Bus Driver

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the PVTC.

Interface Logic

The interface logic contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are as shown in Table 1.

Table 1. PVTC Addressing

A2	A1	A0	READ ($\overline{RD}$ = 0)	WRITE ($\overline{WR}$ = 0)
0	0	0	Interrupt register	Initialization registers[1]
0	0	1	Status register	Command register
0	1	0	Screen start address lower register	Screen start address lower reg.
0	1	1	Screen start address upper register	Screen start address upper reg.
1	0	0	Cursor address lower register	Cursor address lower register
1	0	1	Cursor address upper register	Cursor address upper register
1	1	0	Light pen address lower register	Display pointer address lower reg.
1	1	1	Light pen address upper register	Display pointer address upper reg.

NOTE:
1. There are 11 initialization registers which are accessed sequentially via a single address. The PVTC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR10, the split screen register) is accessed. The pointer then continues to point to the split screen register. Upon power-up or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command.

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
A0 – A2	37 – 39	41 – 43	I	**Address Lines:** Used to select PVTC internal registers for read/write operations and for commands.
D0 – D7	8 – 15	9 – 11, 13 – 17	I/O	**8-Bit Bidirectional 3-State Data Bus:** Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the PVTC take place over this bus. The direction of the transfer is controlled by the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ inputs when the $\overline{\text{CE}}$ input is low. When the $\overline{\text{CE}}$ input is high, the data bus is in the 3-State condition.
$\overline{\text{RD}}$	1	2	I	**Read Strobe:** Active low input. A low on this pin while $\overline{\text{CE}}$ is low causes the contents of the register selected by A0 – A2 to be placed on the data bus. The read cycle begins on the leading (falling) edge of $\overline{\text{RD}}$.
$\overline{\text{WR}}$	3	4	I	**Write Strobe:** Active low input. A low on this pin while $\overline{\text{CE}}$ is also low causes the contents of the data bus to be transferred to the register selected by A0 – A2. The transfer occurs on the trailing (rising) edge of $\overline{\text{WR}}$.
$\overline{\text{CE}}$	2	3	I	**Chip Enable:** Active low input. When low, data transfers between the CPU and the PVTC are enabled on D0 – D7 as controlled by the $\overline{\text{WR}}$, $\overline{\text{RD}}$, and A0 – A2 inputs. When $\overline{\text{CE}}$ is high, the PVTC is effectively isolated from the data bus and D0 – D7 are placed in the 3-State condition.
$\overline{\text{CCLK}}$	16	18	I	**Character Clock:** Timing signal derived from the video dot clock which is used to synchronize the PVTC's timing functions.
HSYNC	19	21	O	**Horizontal Sync:** Active high output which provides video horizontal sync pulses. The timing parameters are programmable.
VSYNC/CSYNC	18	20	O	**Vertical Sync/Composite Sync:** A control bit selects either vertical or composite sync pulses on this active high output. When CSYNC is selected, equalization pulses are included. The timing parameters are programmable.
BLANK	17	19	O	**Blank:** This active high output defines the horizontal and vertical borders of the display. Display control signals which are output on DADD3 through DADD13 are valid on the trailing edge of BLANK.
CURSOR	7	8	O	**Cursor Gate:** This active high output becomes active for a specified number of scan lines when the address contained in the cursor registers match the address output on DADD0 through DADD13. The first and last lines of the cursor and a blink option are programmable.
$\overline{\text{INTR}}$	35	39	O	**Interrupt Request:** Open drain output which supplies an active low interrupt request from any of five maskable sources. This pin is inactive after power-on reset or a master reset command.
LPS	36	40	I	**Light Pen Strobe:** Positive edge-triggered input indicating a light pen hit. Causes the current value of the display address to be strobed into the light pen register.
CTRL1	4	5	I/O	**Handshake Control 1:** In independent mode, provides an active low write data buffer ($\overline{\text{WDB}}$) output which strobes data from the interface latch into the display memory. In transparent and shared modes, this is an active low processor bus request ($\overline{\text{PBREQ}}$) input which indicates that the CPU desires to access the display memory. This pin must be tied high when operating in row buffer mode.
CTRL2	5	6	O	**Handshake Control 2:** In independent mode, provides an active low read data buffer ($\overline{\text{RDB}}$) output which strobes data from the display memory into the interface latch. In transparent and shared modes, this is an active low bus external enable ($\overline{\text{BEXT}}$) output which indicates that the PVTC has relinquished control of the display memory (DADD0 – DADD13 are in the 3-State condition) in response to a CPU bus request. $\overline{\text{BEXT}}$ also goes low in response to a 'display off and float DADD' command. In row buffer mode, it is an active low bus request ($\overline{\text{BREQ}}$) output which halts the CPU during a line DMA.
CTRL3	6	7	O	**Handshake Control 3:** In independent mode, provides the active low buffer chip enable ($\overline{\text{BCE}}$) signal to the display memory. In transparent and shared modes provides an active low bus acknowledge ($\overline{\text{BACK}}$) output which serves as a ready signal to the CPU in response to a processor bus request. In row buffer mode, this is an active high memory bus control

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
				(MBC) output which configures the system for the DMA transfer of one row of character codes from system memory to the row display buffer.
DADD0 - DADD13	34 - 21	38 - 35 33 - 24	O	**Display Address:** Used by the PVTC to address up to 16k of display memory. These outputs are floated at various times depending on the buffer mode. Various control signals are multiplexed on DADD3 thru DADD13 and are valid at the trailing edge of BLANK. These control signals are: **DADD3/LI** **Line Interlace:** Replaces DADD4/LA0 as the least significant line address for the interlaced sync and video applications. A low indicates an even row of an even field or an odd row of an odd field. **DADD4 - DADD7/LA0 - LA3** **Line Address:** Provides the number of the current scan line within each character row. **DADD8/LNZ** **Line Zero:** Asserted before the first scan line in each character row. **DADD9/LPL** **Light Pen Line:** Asserted before the scan line which matches the programmed light pen line position (line 3, 5, 7, or 9). **DADD10/UL** **Underline:** Asserted before the scan line which matches the programmed underline position (lines 0 thru 15). **DADD11/BLINK** **Blink frequency:** Provides an output divided down from the vertical sync rate. **DADD12/ODD** **Odd Field:** Active high signal which is asserted before each scan line of the odd field when interlace is specified. **DADD13/LL** **Last Line:** Asserted before the last scan line of each character row.
V_{CC}	40	44	I	**Power Supply:** +5V ±5% power input.
GND	20	22	I	**Ground:** Signal and power ground input.

Operation Control

The operation control section decodes configuration and operation commands from the CPU and generates appropriate signals to other internal sections to control the overall device operation. It contains the timing and display registers which configure the display format and operating mode, the interrupt logic, and the status register which provides operational feedback to the CPU.

Timing

The timing section contains the counters and decoding logic necessary to generate the monitor timing outputs and to control the display format. These timing parameters are selected by programming of the initialization registers.

Display Control

The display control section generates linear addressing for up to 16k bytes of display memory. Internal comparators limit the portion of the memory which is displayed to programmed values. Additional functions performed in this section include cursor positioning, storage of light pen 'hit' location, and address comparisons required for generation of timing signals and the split screen interrupt.

Buffer Control

The buffer control section generates three signals which control the transfer of data between the CPU and the display buffer memory. Four system configurations requiring four different 'handshaking' schemes are supported. These are described below.

SYSTEM CONFIGURATIONS

Figure 1 illustrates the block diagram of a typical display terminal using the Signetics 2670, 2671, 2672, and 2673/2677 CRT terminal devices. In this system, the CPU examines inputs from the data communications line and the keyboard and places the data to be displayed in the display buffer memory. This buffer is typically a RAM which holds the data for a single or multiple screenload (page) or for a single character row.

The PVTC supports four common system configurations of display buffer memory: the independent, transparent, shared, and row buffer modes. The first three modes utilize a single or multiple page RAM and differ primarily in the means used to transfer display data between the RAM and the CPU. The row buffer mode makes use of a single row buffer (which can be a shift register or a small RAM) that is updated in real-time to contain the appropriate display data.

The user programs bits 0 and 1 of IR0 to select the mode best suited for the system environment. The CNTRL1 - 3 outputs perform different functions for each mode and are named accordingly in the description of each mode.

Independent Mode

The CPU to RAM interface configuration for this mode is illustrated in Figure 2. Transfer of data between the CPU and display memory is accomplished via a bidirectional latched port and is controlled by the signals read data buffer ($\overline{RDB}$), write data buffer ($\overline{WDB}$), and

Programmable Video Timing Controller (PVTC) SCN2672

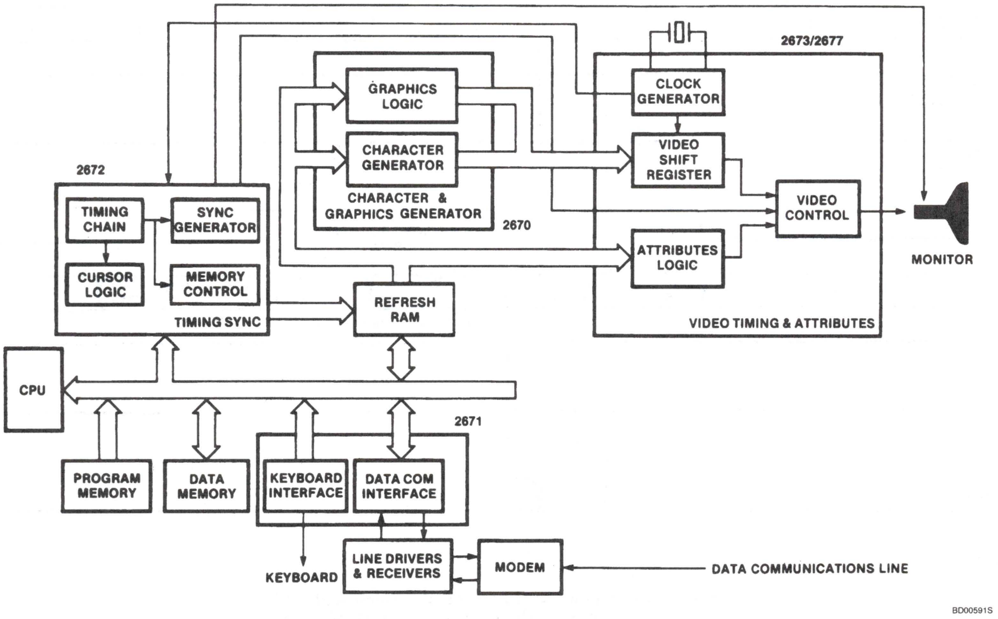

2673/2677
CLOCK GENERATOR
GRAPHICS LOGIC
CHARACTER GENERATOR
CHARACTER & GRAPHICS GENERATOR
2670
VIDEO SHIFT REGISTER
VIDEO CONTROL
ATTRIBUTES LOGIC
MONITOR
VIDEO TIMING & ATTRIBUTES
2672
TIMING CHAIN
SYNC GENERATOR
CURSOR LOGIC
MEMORY CONTROL
TIMING SYNC
REFRESH RAM
CPU
2671
PROGRAM MEMORY
DATA MEMORY
KEYBOARD INTERFACE
DATA COM INTERFACE
KEYBOARD
LINE DRIVERS & RECEIVERS
MODEM
DATA COMMUNICATIONS LINE
BD00591S

Figure 1. CRT Terminal Block Diagram

buffer chip enable ($\overline{BCE}$). This mode provides a non-contention type of operation that does not require address multiplexers. The CPU does not address the memory directly — the read or write operation is performed at the address contained in the cursor address register or the pointer address register as specified by the CPU. The PVTC enacts the data transfers during blanking intervals in order to prevent visual disturbances of the displayed data.

For a data buffer write command, the $\overline{WDB}$ signal will go active on the rising edge of character clock ($\overline{CCLK}$) and will remain so until the next $\overline{CCLK}$ rising edge. $\overline{BCE}$ is always in the active state except before and after a $\overline{WDB}$ command. When a write has been executed, $\overline{BCE}$ becomes inactive on the falling edge of $\overline{CCLK}$. When the write occurs ($\overline{WDB}$ active) on the next rising edge, $\overline{BCE}$ also becomes active. $\overline{BCE}$ and $\overline{WDB}$ both become inactive on the rising edge of $\overline{CCLK}$. $\overline{BCE}$ will then return to the active state on the next falling edge if there is no other write command to be executed.

The CPU manages the data transfers by supplying commands to the PVTC. The commands used are:

1. Read/Write at pointer address.
2. Read/Write at cursor address (with optional increment of address).
3. Write from cursor address to pointer address.

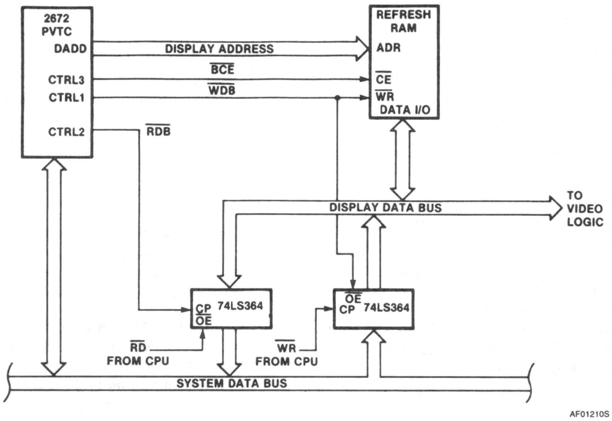
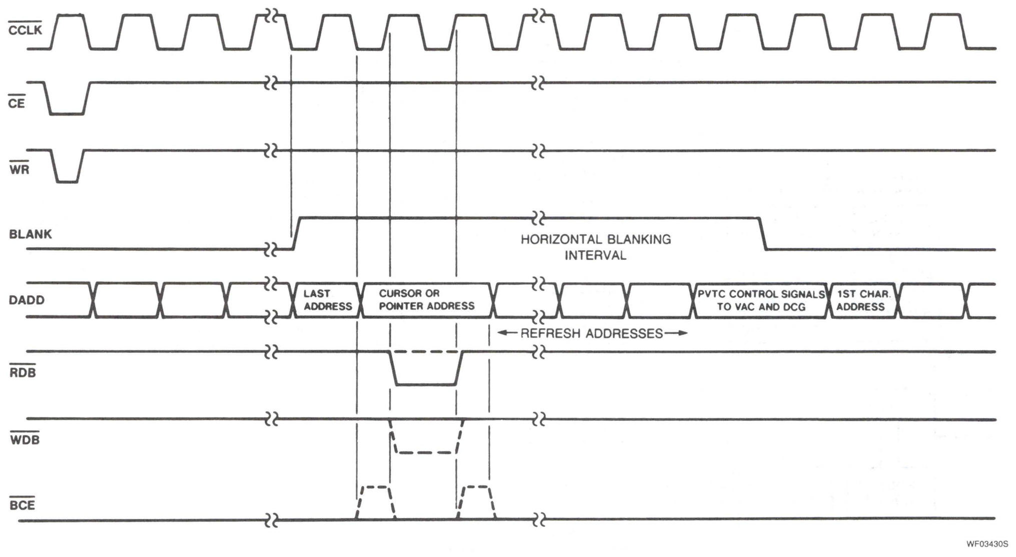

2672 PVTC
DADD
CTRL3
CTRL1
CTRL2
DISPLAY ADDRESS
BCE
WDB
RDB
REFRESH RAM
ADR
CE
WR
DATA I/O
DISPLAY DATA BUS
TO VIDEO LOGIC
CP OE 74LS364
OE CP 74LS364
RD FROM CPU
WR FROM CPU
SYSTEM DATA BUS
AF01210S

Figure 2. Independent Buffer Mode Configuration

The operational sequence for a write operation is:

1. CPU checks RDFLG status bit to assure that any previous operation has been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes address into cursor or pointer registers.


CCLK
CE
WR
BLANK
HORIZONTAL BLANKING INTERVAL
DADD
LAST ADDRESS
CURSOR OR POINTER ADDRESS
PVTC CONTROL SIGNALS TO VAC AND DCG
1ST CHAR. ADDRESS
← REFRESH ADDRESSES →
RDB
WDB
BCE
WF03430S

NOTE:
Write waveforms shown in dotted lines.

Figure 3. Read/Write at Cursor/Pointer Command Timing (Command Received During Active Display Window)

4. CPU issues 'write at cursor with/without increment' or 'write at pointer' command.
5. PVTC generates control signals and outputs specified address to perform requested operation. Data is copied from the interface latch into the memory.
6. PVTC sets RDFLG status to indicate that the write is completed.

Similarly, a read operation proceeds as follows:

1. Steps 1 and 3 as above.
2. CPU issues 'read at cursor with/without increment' or 'read at pointer' command.
3. PVTC generates control signals and outputs specified address to perform requested operation. Data is copied from memory to the interface latch and PVTC sets RDFLG status to indicate that the read is completed.
4. CPU checks RDFLG status to see if operation is completed.
5. CPU reads data from interface latch.

Loading the same data into a block of display memory is accomplished via the 'write from cursor to pointer' command:

1. CPU checks RDFLG status bit to assure that any previous operation has been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes beginning address of memory block into cursor address register and ending address of block into pointer address register.

   issues 'write from cursor to pointer' command.
5. PVTC generates control signals and outputs block addresses to copy data from the interface latch into the specified block of memory.
6. PVTC sets RDFLG status to indicate that the block write is completed.

Similar sequences can be implemented on an interrupt driven basis using the READY interrupt output to advise the CPU that a previously requested command has been completed.

Two timing sequences are possible for the 'read/write at cursor/pointer' commands. If the command is given during the active display window (defined as first scan line of the first character row to the last scan line of the last character row), the operation takes place during the next horizontal blanking interval, as illustrated in Figure 3. If the command is given during the vertical blanking interval, or while the display has been commanded blanked, the operation takes place immediately. In the latter case, the execution time for the command is approximately one microsecond plus six (6) character clocks (see Figure 4).

Timing for the 'write from cursor to pointer' operation is shown in Figure 5. The BLANK output is asserted automatically and remains asserted until the horizontal retrace interval following completion of the command. The memory is filled at a rate of one location per two character times, plus a small amount of overhead.

Shared And Transparent Buffer Modes

In these modes the display buffer RAM is a part of the CPU memory domain and is addressed directly by the CPU. Both modes use the same hardware configuration with the CPU accessing the display buffer via 3-State drivers (see Figure 6). The processor bus request ($\overline{\text{PBREQ}}$) control signal informs the PVTC that the CPU is requesting access to the display buffer. In response to this request, the PVTC raises bus acknowledge ($\overline{\text{BACK}}$) until its bus external ($\overline{\text{BEXT}}$) output has freed the display address and data busses for CPU access. $\overline{\text{BACK}}$, which can be used as a 'hold' input to the CPU, is then lowered to indicate that the CPU can access the buffer.

In transparent mode, the PVTC delays the granting of the buffer to the CPU until a vertical or horizontal blanking interval, thereby causing minimum disturbance of the display. In shared mode, the PVTC will blank the display and grant immediate access to the CPU. Timing for these modes is illustrated in Figures 7, 8, and 9.

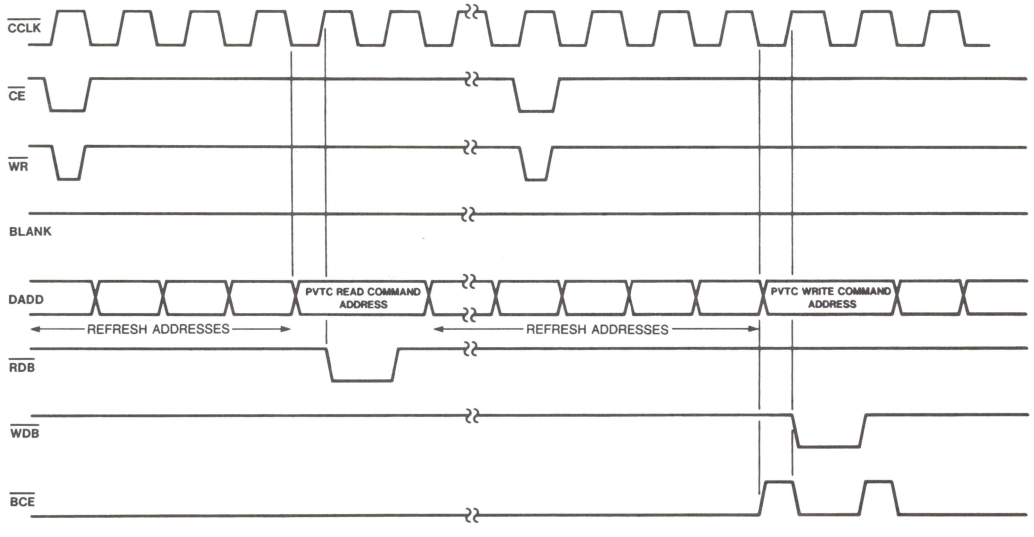

CCLK
CE
WR
BLANK
DADD
PVTC READ COMMAND ADDRESS
PVTC WRITE COMMAND ADDRESS
REFRESH ADDRESSES
REFRESH ADDRESSES
RDB
WDB
BCE
WF03440S

Figure 4. Read/Write at Cursor/Pointer Command Timing (Command Received While Display is Blanked)

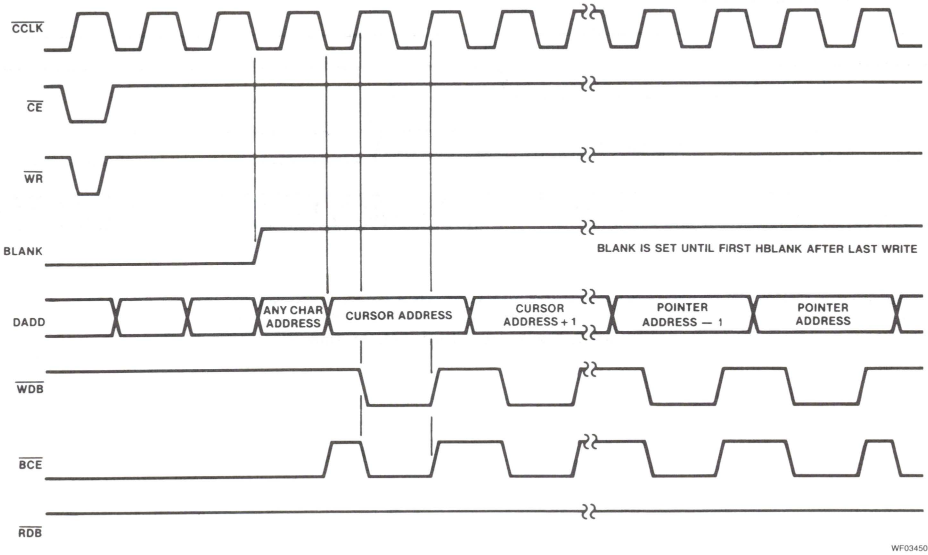

CCLK
CE
WR
BLANK
BLANK IS SET UNTIL FIRST HBLANK AFTER LAST WRITE
DADD
ANY CHAR ADDRESS
CURSOR ADDRESS
CURSOR ADDRESS + 1
POINTER ADDRESS − 1
POINTER ADDRESS
WDB
BCE
RDB
WF03450S

Figure 5. Write From Cursor to Pointer Command Timing

Programmable Video Timing Controller (PVTC) | SCN2672

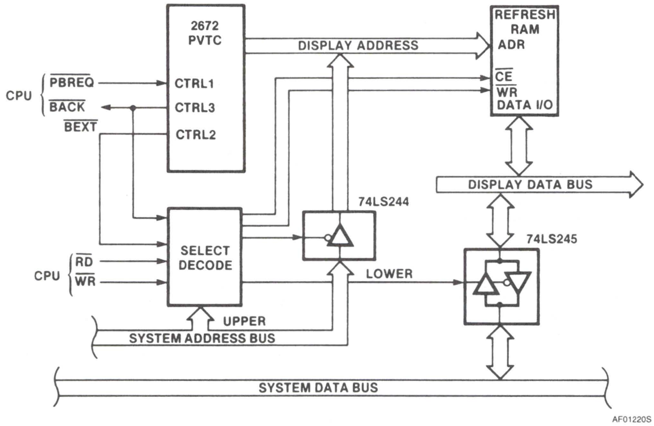
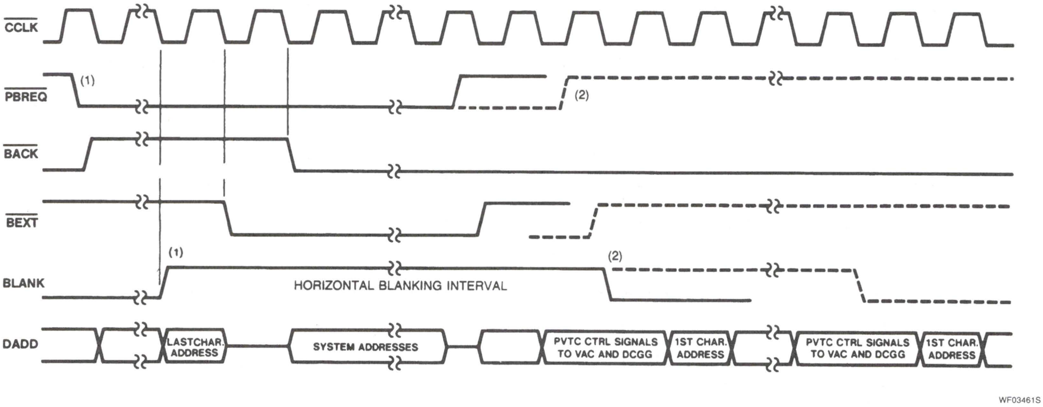

2672
PVTC
DISPLAY ADDRESS
REFRESH RAM
ADR
CE
WR
DATA I/O
CPU
PBREQ
BACK
BEXT
CTRL1
CTRL3
CTRL2
DISPLAY DATA BUS
74LS244
SELECT DECODE
74LS245
CPU
RD
WR
LOWER
UPPER
SYSTEM ADDRESS BUS
SYSTEM DATA BUS
AF01220S

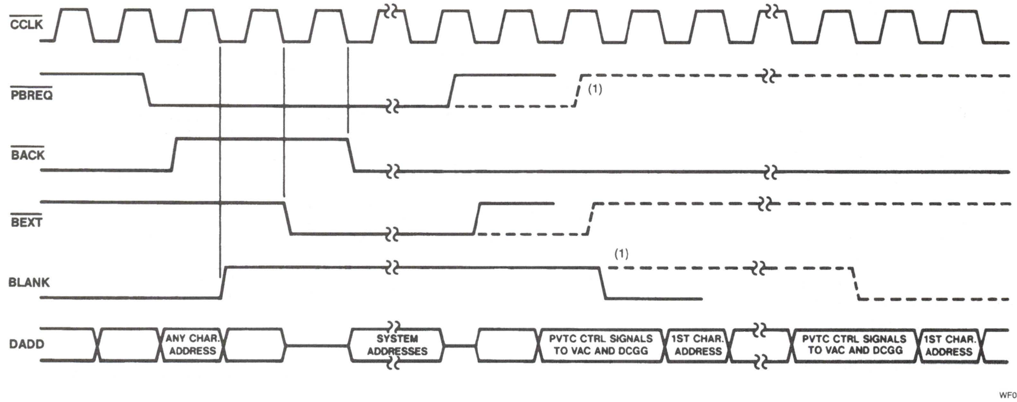
Figure 6. PVTC Shared or Transparent Buffer Modes


CCLK
PBREQ
(1)
(2)
BACK
BEXT
BLANK
(1)
(2)
HORIZONTAL BLANKING INTERVAL
DADD
LASTCHAR. ADDRESS
SYSTEM ADDRESSES
PVTC CTRL SIGNALS TO VAC AND DCGG
1ST CHAR. ADDRESS
PVTC CTRL SIGNALS TO VAC AND DCGG
1ST CHAR. ADDRESS
WF03461S

NOTES:

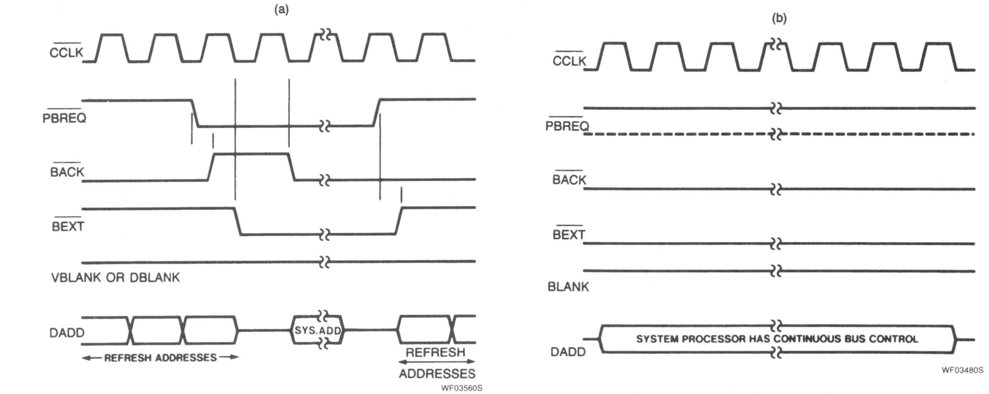
1. $\overline{\text{PBREQ}}$ must be asserted prior to the rising edge of BLANK in order for sequence to begin during that blanking period.
2. If $\overline{\text{PBREQ}}$ is negated after the next to last $\overline{\text{CCLK}}$ of the horizontal blanking interval, the next scan line will also be blanked.

Figure 7. Transparent Buffer Mode Timing


CCLK
PBREQ
BACK
BEXT
BLANK
DADD
(1)
(1)
ANY CHAR. ADDRESS
SYSTEM ADDRESSES
PVTC CTRL SIGNALS TO VAC AND DCGG
1ST CHAR. ADDRESS
PVTC CTRL SIGNALS TO VAC AND DCGG
1ST CHAR. ADDRESS
WF03471S

NOTE:
1. If $\overline{\text{PBREQ}}$ is negated after the next to last $\overline{\text{CCLK}}$ of the horizontal blanking interval, the next scan line will also be blanked.

Figure 8. Shared Buffer Mode Timing


(a)
CCLK
PBREQ
BACK
BEXT
VBLANK OR DBLANK
DADD
REFRESH ADDRESSES
SYS.ADD
REFRESH ADDRESSES
WF03560S
(b)
CCLK
PBREQ
BACK
BEXT
BLANK
DADD
SYSTEM PROCESSOR HAS CONTINUOUS BUS CONTROL
WF03480S

a) During Vertical Blank or After 'Display Off' Command

b) After 'Display Off and 3-State' Command

Figure 9. Shared and Transparent Mode Timing

Row Buffer Mode

Figures 10 and 11 show the timing and a typical hardware implementation for the row buffer mode. During the first scan line (line 0) of each character row, the PVTC halts the CPU and DMAs the next row of character data from the system memory to the row buffer memory. The PVTC then releases the CPU and displays the row buffer data for the programmed number of scan lines. The bus request control (BREQ) signal informs the CPU that character addresses and the memory bus control (MBC) signal will start at the next falling edge of BLANK. The CPU must release the address and data buses before this time to prevent bus contention. After the row of character data is transferred to the row buffer RAM, BREQ returns high to grant memory control back to the CPU.

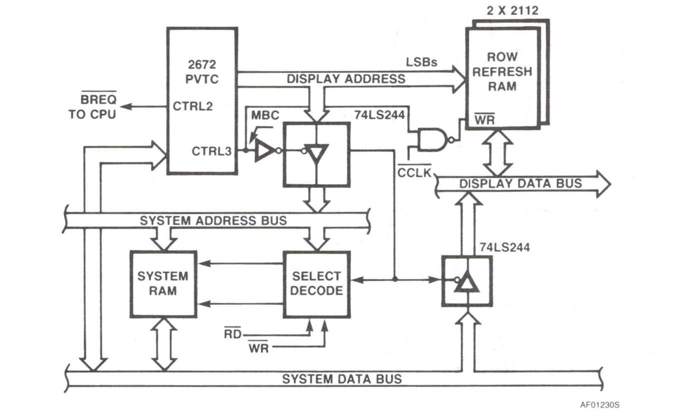


Figure 10. Row Buffer Mode Configuration

OPERATION

After power is applied, the PVTC will be in an inactive state. Two consecutive 'master reset' commands are necessary to release this circuitry and ready the PVTC for operation. Two register groups exist within the PVTC: the initialization registers and the display control registers. The initialization registers select the system configuration, monitor timing, cursor shape, display memory domain, and screen format. These are loaded first and normally require no modification except for certain special visual effects. The display control registers specify the memory address of the base character (upper left corner of screen), the cursor position, and the pointer address for independent memory access mode. These usually require modification during operation.

After initial loading of the two register groups, the PVTC is ready to control the monitor screen. Prior to executing the PVTC commands which turn on the display and cursor, the user should load the display memory with the first data to be displayed. During operation, the PVTC will sequentially address the display memory within the limits programmed into its registers. The memory outputs character codes to the system character and graphics generation logic, where they are converted to the serial video stream necessary to display the data on the CRT. The user effects changes to the display by modifying the contents of the display memory, the PVTC display control and command registers, and the initialization registers, if required. Interrupts and status conditions generated by the PVTC supply the 'handshaking' information necessary for the CPU to effect the display changes in the proper time frame.

Initialization Registers

There are 11 initialization registers (IR0 – IR10) which are accessed sequentially via a single address. The PVTC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR10, the split screen register) is accessed. The pointer then continues to point to the split screen register. Upon power-up or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command. These registers are write only and are used to specify parameters such as the system configuration, display format, cursor shape, and monitor timing. Register formats are shown in Table 2.

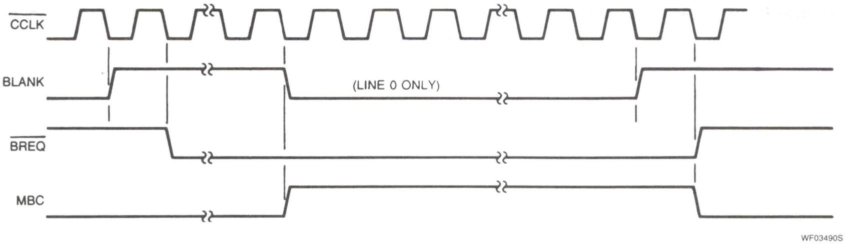


Figure 11. Row Buffer Mode Timing

Table 2. Initialization Register Bit Formats

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR0	NOT USED	**SCAN LINES PER CHARACTER ROW**				**SYNC SELECT**	**BUFFER MODE SELECT**	
		NON-INTERLACED		**INTERLACED**				
		0000 = 1 LINE 0001 = 2 LINES 0010 = 3 LINES • • 1110 = 15 LINES 1111 = 16 LINES		0000 = UNDEFINED 0001 = 5 LINES 0010 = 7 LINES • • 1110 = 31 LINES 1111 = UNDEFINED		0 = VSYNC 1 = CSYNC	00 = INDEPENDENT 01 = TRANSPARENT 10 = SHARED 11 = ROW	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR1	**INTERLACE ENABLE**	**EQUALIZING CONSTANT**						
	0 = NON-INT 1 = INTER.	00000000 = 1 CCLK 00000001 = 2 CCLK • • 11111110 = 127 CCLK 11111111 = 128 CCLK			CALCULATED FROM: $EC = 0.5(H_{ACT} + H_{FP} + H_{SYNC} + H_{BP}) - 2(H_{SYNC})$			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR2	NOT USED	**HORIZONTAL SYNC WIDTH**				**HORIZONTAL BACK PORCH**		
		0000 = 2 CCLK 0001 = 4 CCLK • • 1110 = 30 CCLK 1111 = 32 CCLK				000 = 1 CCLK 001 = 5 CCLK • • 110 = 25 CCLK 111 = 29 CCLK		

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR3	**VERTICAL FRONT PORCH**			**VERTICAL BACK PORCH**				
	000 = 4 SCAN LINES 001 = 8 SCAN LINES • • 110 = 28 SCAN LINES 111 = 32 SCAN LINES			00000 = 4 SCAN LINES 00001 = 6 SCAN LINES • • 11110 = 64 SCAN LINES 11111 = 66 SCAN LINES				

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR4	**CHARACTER BLINK RATE**	**ACTIVE CHARACTER ROWS PER SCREEN (NOTE 1)**						
	0 = 1/16 VSYNC 1 = 1/32 VSYNC	00000000 = 1 ROW 00000001 = 2 ROWS • • 11111110 = 127 ROWS 11111111 = 128 ROWS						

NOTE:
In interlace mode with odd total character rows per screen the last character row will be the programmed scan lines per character row minus one.

Table 2. Initialization Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	ACTIVE CHARACTERS PER ROW							
IR5	00000010 = 3 CHARACTERS 00000011 = 4 CHARACTERS • • 11111110 = 255 CHARACTERS 11111111 = 256 CHARACTERS							

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	FIRST LINE OF CURSOR				LAST LINE OF CURSOR			
IR6	0000 = SCAN LINE 0 0001 = SCAN LINE 1 • • 1110 = SCAN LINE 14 1111 = SCAN LINE 15				0000 = SCAN LINE 0 0001 = SCAN LINE 1 • • 1110 = SCAN LINE 14 1111 = SCAN LINE 15			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	LIGHT PEN LINE		CURSOR BLINK	DOUBLE HEIGHT CHAR.	UNDERLINE POSITION			
IR7	00 = SCAN LINE 3 01 = SCAN LINE 5 10 = SCAN LINE 7 11 = SCAN LINE 9		0= NO 1= YES	0= NO 1= YES	0000 = SCAN LINE 0 0001 = SCAN LINE 1 • • 1110 = SCAN LINE 14 1111 = SCAN LINE 15			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	DISPLAY BUFFER FIRST ADDRESS LSB'S							
IR8	H'000' = 0 H'001' = 1 • • H'FFE' = 4,094 H'FFF' = 4,095 NOTE: MSB'S ARE IN IR9[3:0]							

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	DISPLAY BUFFER LAST ADDRESS				DISPLAY BUFFER FIRST ADDRESS MSB'S			
IR9	0000 = 1,023 0001 = 2,047 • • 1110 = 15,359 1111 = 16,383				SEE IR8			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	CURSOR BLINK RATE	SPLIT SCREEN INTERRUPT ROW						
IR10	0 = 1/16 VSYNC 1 = 1/32 VSYNC	00000000 = ROW 0 00000001 = ROW 1 • • 11111110 = ROW 126 11111111 = ROW 127						

IR0[6:3] — Scan Lines Per Character Row
Both interlaced and non-interlaced scanning are supported by the PVTC. For interlaced mode, two different formats can be implemented, depending on the interconnection between the PVTC and the character generator (see IR1[7]). This field defines the number of scan lines used to compose a character row for each technique. As scanning occurs, the scan line count is output on the LA0 – LA3 and LI pins.

IR0[2] — VS/CS Enable
This bit selects either vertical sync pulses or composite sync pulses on the VSYNC/CSYNC output (pin 18). The composite sync waveform conforms to EIA RS170 standards, with the vertical interval composed of six equalizing pulses, six vertical sync pulses, and six more equalizing pulses.

IR0[1:0] — Buffer Mode Select
Four buffer memory modes may be selectively enabled to accommodate the desired system configuration. See System Configurations.

IR1[7] — Interface Enable
Specifies interlaced or noninterlaced timing operation. Two modes of interlaced operation are available, depending on whether LA0 – LA3 or LI, LA0 – LA2 are used as the line address for the character generator. The resulting displays are shown in Figure 12.

For 'interlaced sync' operation, the same information is displayed in both odd and even fields, resulting in enhanced readability. The PVTC outputs successive line numbers in ascending order on the LA0 – LA3 lines, one per scan line for each field.

The 'interlaced sync and video' format doubles the character density on the screen. The PVTC outputs successive line numbers in ascending order on the LI, LA0 – LA2 lines, one per scan line for each field, but alternates beginning the count with even and odd line numbers. In the interlaced sync and video mode, the number of scan lines per character row is always odd. Assume that the first character row is row 0 (even). When in the odd field, the scan line numbers being displayed are even for even character rows and odd for odd character rows. When in the even field, the scan line numbers being displayed are odd for even character rows and even for odd character rows (see Figure 12c). This provides balanced beam currents in the odd and even fields, thus minimizing character variations due to different loading of the CRT anode supply between fields.

IR1[6:0] — Equalizing Constant
This field indirectly defines the horizontal front porch and is used internally to generate the equalizing pulses for the RS170 compatible CSYNC. The value for this field is the total number of character clocks ($\overline{\text{CCLK}}$) during a horizontal line period divided by two, minus two times the number of character clocks in the horizontal sync pulse:

$$EC = \frac{H_{ACT} + H_{FP} + H_{SYNC} + H_{BP}}{2} - 2(H_{SYNC})$$

The definition of the individual parameters is illustrated in Figure 13. The minimum value of H_{FP} is two character clocks.

Note that when using the 2673/2677 VAC, the blank pulse is delayed three CCLKs relative to the HSYNC pulse. Because of this delay, the actual HFP and HBP values will be different from the values programmed into the PVTC. The actual HFP will be decreased by 3 character clocks. The actual HBP will be increased by 3 character clocks.

IR2[6:3] — Horizontal Sync Pulse Width
This field specifies the width of the HSYNC pulse in $\overline{\text{CCLK}}$ periods.

IR2[2:0] — Horizontal Back Porch
This field defines the number of $\overline{\text{CCLKs}}$ between the trailing edge of HSYNC and the trailing edge of BLANK.

IR3[7:5] — Vertical Front Porch
Programs the number of scan line periods between the rising edges of BLANK and VSYNC during a vertical retrace interval. The width of the VSYNC pulse is fixed at three scan lines.

IR3[4:0] — Vertical Back Porch
This field determines the number of scan line periods between the falling edges of the VSYNC and BLANK outputs.

IR4[7] — Character Blink Rate
Specifies the frequency for the character blink attribute timing. The blink rate can be specified as 1/16 or 1/32 of the vertical field rate. The timing signal has a duty cycle of 75% and is multiplexed onto the DADD11/BLINK output at the falling edge of each BLANK.

IR4[6:0] — Character Rows Per Screen
This field defines the number of character rows to be displayed. This value multiplied by the scan lines per character row, plus the vertical front and back porch values, and the vertical sync pulse width (three scan lines) is the vertical scan period in scan lines.

IR5[7:0] — Active Characters Per Row
This field determines the number of characters to be displayed on each row of the CRT screen. The sum of this value, the horizontal front porch, the horizontal sync width, and the horizontal back porch is the horizontal scan period in $\overline{\text{CCLKs}}$.

IR6[7:4], IR6[3:0] — First and Last Scan Line of Cursor
These two fields specify the height and position of the cursor on the character block. The 'first' line is the topmost line when scanning from the top to the bottom of the screen. The value of the first line of cursor must be less than the last line of cursor value.

IR7[7:6] — Light Pen Line Position
This field defines which of four scan lines of the character row will be used for the light pen strike-through attribute by the 2673/2677 VAC. The timing signal is multiplexed onto the DADD9/LPL output during the falling edge of BLANK.

IR7[5] — Cursor Blink Enable
This bit controls whether or not the cursor output pin will be blinked at the selected rate (IR10[7]). The blink duty cycle for the cursor is 50%.

IR7[4] — Double Height Character Row Enable
If enabled, the scan line count will be repeated twice in succession, causing the height of the character row to double. This bit can be changed at any time but will only become effective at the beginning of the character row following the time it is changed. This allows selected character rows to be of double height. The split screen interrupt can be used to notify the CPU when to effectuate changes to this bit. For each double height row which replaces a normal row, one row count should be subtracted from the 'character rows per screen' field (IR4) to maintain the same total number of scan lines per field.

IR7[3:0] — Underline Position
This field defines which scan line of the character row will be used for the underline attribute by the 2673/2677 VAC. The timing signal is multiplexed onto the DADD10/UL output during the falling edge of BLANK.

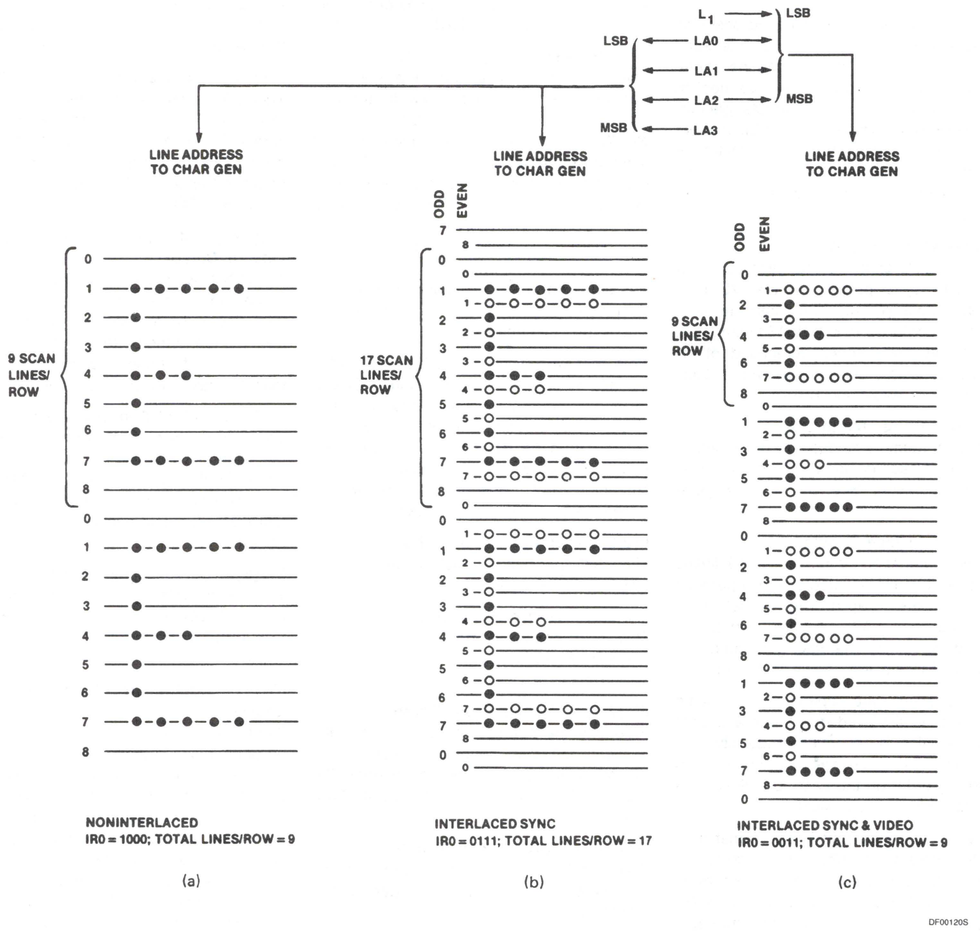

L1
LSB
LA0
LA1
LA2
LA3
LSB
MSB
MSB
LINE ADDRESS TO CHAR GEN
LINE ADDRESS TO CHAR GEN
LINE ADDRESS TO CHAR GEN
ODD
EVEN
ODD
EVEN
9 SCAN LINES/ ROW
17 SCAN LINES/ ROW
9 SCAN LINES/ ROW
NONINTERLACED
IR0 = 1000; TOTAL LINES/ROW = 9
INTERLACED SYNC
IR0 = 0111; TOTAL LINES/ROW = 17
INTERLACED SYNC & VIDEO
IR0 = 0011; TOTAL LINES/ROW = 9
(a)
(b)
(c)
DF00120S

Figure 12. Interlaced Display Modes

IR9[3:0], IR8[7:0] — Display Buffer First Address

IR9[7:4] — Display Buffer Last Address

These two fields define the area within the buffer memory where the display data will reside. When the data at the 'display buffer last address' is displayed, the PVTC will wraparound and obtain the data to be displayed at the next screen position from the 'display buffer first address'. If 'last address' is the end of a character row and a new screen start address has been loaded into the screen start register, or if 'last address' is the last character position of the screen, the next data is obtained from the address contained in the screen start register.

Note that there is no restriction in displaying data from other areas of the addressable memory. Normally, the area between these two bounds is used for data which can be overwritten (e.g., as a result of scrolling), while data that is not to be overwritten would be contained outside these bounds and accessed by means of the split screen interrupt feature of the PVTC.

IR10[7] — Cursor Blink Rate

The cursor blink rate can be specified at 1/16 or 1/32 of the vertical scan frequency. Blink is effective only if blink is enabled by IR7[5].

IR10[6:0] — Split Screen Interrupt

The split screen interrupt can be used to provide special screen effects such as a row of double height characters or to change the normal addressing sequence of the display memory. The contents of this field are compared, in real-time, to the current character row number. Upon a match, the PVTC sets the split screen status bit, and issues an interrupt request if so programmed. The status change/interrupt request is made at the beginning of scan line zero of the split screen character row.

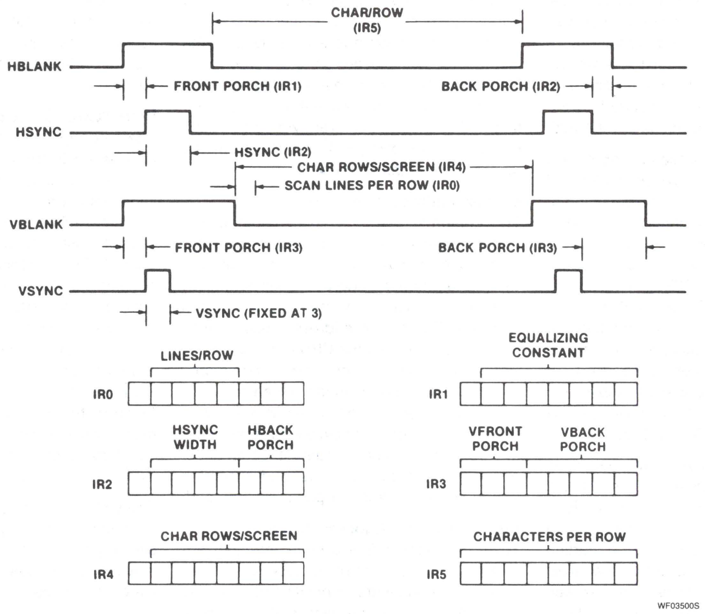


Figure 13. Horizontal and Vertical Timing

Timing Considerations

Normally, the contents of the initialization registers are not changed during operation. However, this may be necessary to implement special display features such as multiple cursors, smooth scrolling, horizontal scrolling, and double height character rows. Table 3 describes timing details for these registers which should be considered when implementing these features.

Table 3. Timing Considerations

PARAMETER	TIMING CONSIDERATIONS
First line of cursor Last line of cursor Light pen line Underline	These parameters must be established at a minimum of two character times prior to their occurrence.
Double height characters	Set/reset during the character row prior to the affected row.
Cursor blink Cursor blink rate Character blank rate	New values become effective within one field after values are changed
Split screen interrupt row	Change anytime prior to line zero of desired row
Character rows per screen	Change only during vertical blanking period
Vertical front porch	Change prior to first line of V_{FP}
Vertical back porch	Change prior to fourth line after V_{SYNC}
Screen start register	Change prior to the horizontal blanking interval of the last line of character row prior to the affected row.

Display Control Registers

There are nine registers in this group, each with an individual address. Their formats are illustrated in Table 4. The command register is used to invoke one of 16 possible PVTC commands as described in the COMMANDS section of this data sheet. The remaining registers in the group store address values which specify the cursor and buffer pointer locations, the location of the first character to be displayed on the screen, and the location of a light pen 'hit'. With the exception of the light pen register, the user initializes these registers after powering on the system and changes their values to control the data which is displayed.

Screen Start Registers

The screen start registers contain the address of the first character of the first row (upper left corner of the active display). At the beginning of the first scan line of the first row, this address is transferred to the row start register (RSR) and into the memory address counter (MAC). The counter is then advanced sequentially at the character rate the number of times programmed into the active characters per row register (IR5), thus reaching the address of the last character of the row plus one. At the beginning of each subsequent scan line of the first row, the MAC is reloaded from the RSR and the above sequence is repeated. At the end of the last scan line of the first row, the contents of the MAC are loaded into the RSR to serve as the starting memory address for the second character row. This process is repeated for the programmed number of rows per screen. Thus, the data in the display memory is displayed sequentially starting from the address contained in the screen start register. After the ensuing vertical retrace interval the contents of the screen registers are reloaded into the RSR and MAC and the process is repeated.

The sequential operation described above will be modified upon the occurrence of either of two events. First, if during the incrementing of the memory address counter the 'display buffer last address' (IR9[7:4]) is reached, the MAC will be loaded from the 'display buffer first address' register (IR9[3:0], IR8[7:0]), at the next character clock. Sequential operation will then resume starting from this address. This wraparound operation allows portions of the display buffer to be used for purposes other than storage of displayable data and is completely automatic without any CPU intervention (see Figure 14a).

Second, the sequential row to row addressing can also be modified under CPU control. If the contents of the screen start register (upper, lower, or both) are changed during any character row (say row 'n'), the starting address of the next character row (row 'n + 1') will be the new value of the screen start register and addressing will continue sequentially from there. This allows features such as split screen operation, partial scroll, or status line display to be implemented. The split screen interrupt feature of the PVTC is useful in controlling this type of operation. Note that in order to obtain the correct screen display, the screen start register must be reloaded with the original value prior to the end of the vertical retrace. See Figure 14b.

Refresh Addressing

During vertical blanking the address counter operation is modified by stopping the automatic load of the contents of the RSR into the counter, thereby allowing the address outputs to free-run. This allows dynamic memory refresh to occur during the vertical retrace interval. The refresh addressing starts at the last address displayed on the screen and increments by one for each character clock during the retrace interval. If the display buffer last address is encountered, wraparound will occur and refreshing will continue from the display buffer first address.

Cursor Address Registers

The contents of these registers define the buffer memory address of the cursor. If enabled, the cursor output will be asserted when the memory address counter (MAC) matches the value of the cursor address registers. The cursor address registers may be read or written by the CPU or incremented via the 'increment cursor address' command. In independent buffer mode, these registers define a buffer memory address for PVTC controlled access in response to 'read/write at cursor with/without increment' commands, or the first address to be used in executing the 'write from cursor to pointer' command.

Display Pointer Address Registers

These registers define a buffer memory address for PVTC controlled accesses in response to 'read/write at pointer' commands. They also define the last buffer memory address to be written for the 'write from cursor to pointer' command.

Light Pen Address Registers

If the light pen input is enabled, these registers are used to store the current character address upon receipt of a light pen strobe input. Several sources of delay between the display of a character upon the screen and the receipt of a light pen hit can be expected to exist in a system environment. These delays include address pipelining in the character generation circuits, delays in the video generation circuits, and delays in the light detection circuitry itself. These delays cause the value stored in the light pen register to differ from the actual address of the character at which the light pen hit actually was detected. Software must be used to correct this condition.

Interrupt/Status Registers

The interrupt and status registers provide information to the CPU to allow it to interact with the PVTC to effect desired changes to implement various display operations. The interrupt register provides information on five possible interrupting conditions, as shown in Table 5. These conditions may be selectively enabled or disabled (masked) from causing interrupts by certain PVTC commands. An interrupt condition which is enabled (mask bit equal to one) will cause the $\overline{\text{INTR}}$ output to be asserted and will cause the corresponding bit in the interrupt register to be set-upon occurrence of the interrupting condition. An interrupt condition which is disabled (mask bit equal to zero) has no effect on either the $\overline{\text{INTR}}$ output or the interrupt register.

The status register provides six bits of status information: the five possible interrupting conditions plus the RDFLG bit. For this register, however, the contents are not affected by the state of the mask bits.

Descriptions of each interrupt/status register bit follow. Unless otherwise indicated, a bit, once set, will remain set until reset by the CPU by issuing a 'reset interrupt/status bits' command. The bits are also reset by a 'master reset' command and upon power-up. This bit is set to a one upon a master reset.

SR[5] — RDFLG

This bit is present in the status register only. A zero indicates that the PVTC is currently executing the previously issued command. A one indicates that the PVTC is ready to accept a new command.

Table 4. Display Control Register Formats

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
COMMAND CODE							
See COMMAND section for command codes							

COMMAND REGISTER (Write only)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UPPER REGISTER							
Not used		MSBs					

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOWER REGISTER (LSBs)							
H'0000' = 0 H'0001' = 1 H'3FFE' = 16,382 H'3FFF' = 16,383 Note: MSBs are in Upper Register [5:0]							

SCREEN START REGISTERS	(READ AND WRITE)
CURSOR ADDRESS REGISTERS	(READ AND WRITE)
POINTER ADDRESS REGISTER	(WRITE ONLY)
LIGHT PEN ADDRESS REGISTER	(READ ONLY)

Table 5. Interrupt and Status Register Format

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
		RDFLG	VBLANK	LINE ZERO	SPLIT SCREEN	READY	LIGHT PEN
Not used always read as 0		0 = Busy 1 = Ready *	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = Busy 1 = Ready	0 = No 1 = Yes

NOTE:
*Status register only. Always 0 when reading interrupt register.

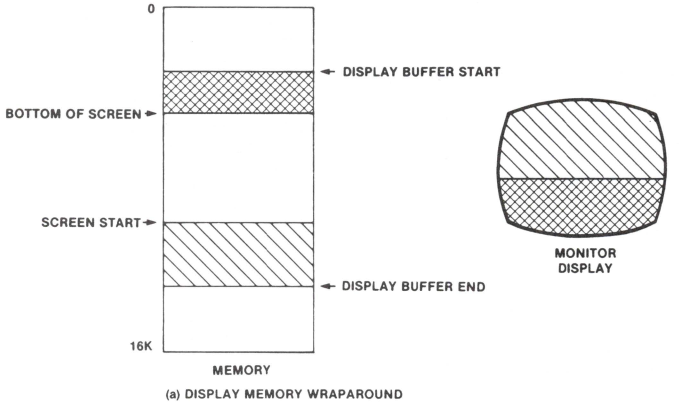
I/SR[4] — VBLANK
Indicates the beginning of a vertical blanking interval. Is set to a one at the beginning of the first scan line of the vertical front porch.

I/SR[3] — Line Zero
Is set to a one at the beginning of the first scan line (line 0) of each active character row.

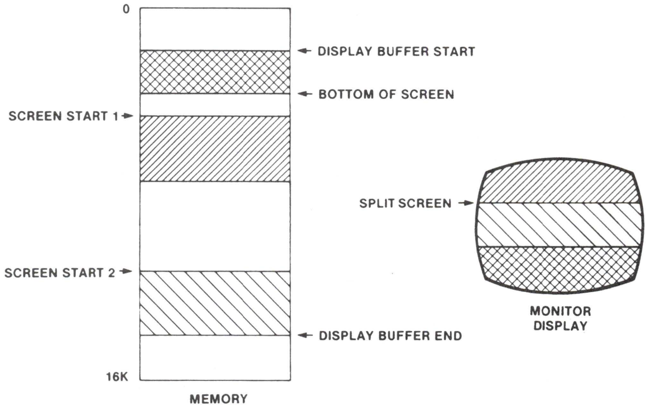
I/SR[2] — Split Screen
This bit is set when a match occurs between the current character row number and the value contained in the split screen interrupts register, IR10[6:0]. The equality condition is only checked at the beginning of line zero of each character row. This bit is reset when either of the screen start registers is loaded by the CPU.

I/SR[1] — Ready
Certain PVTC commands affect the display and may require the PVTC to wait for a blanking interval before enacting the command. This bit is set to one when execution of the command has been completed. No command should be invoked until the prior command is completed. This bit is set to a zero upon a master reset.

I/SR[0] — Light Pen
A one indicates that a light pen hit has occurred and that the contents of the light pen register have been updated. This bit will be reset when either of the light pen registers is read.



(a) DISPLAY MEMORY WRAPAROUND



(b) DISPLAY MEMORY SPLIT SCREEN WITH WRAPAROUND

DF00080S

Figure 14. Display Addressing Operation

COMMANDS

The PVTC commands are divided into two classes: the instantaneous commands, which are executed immediately after they are invoked, and the delayed commands which may need to wait for a blanking interval prior to their execution. Command formats are shown in Table 6. The commands are asserted by performing a write operation to the command register with the appropriate bit pattern as the data byte.

Instantaneous Commands

The instantaneous commands are executed immediately after the trailing edge of the $\overline{WR}$ pulse during which the command is issued. These commands do not affect the state of the RDFLG or READY interrupt/status bits. However, a command should not be invoked if the RDFLG bit is low.

Master Reset
This command initializes the PVTC and may be invoked at any time to return the PVTC to its initial state. Upon power-up, two successive master reset commands must be applied to release the PVTC's internal power on circuits. In transparent and shared buffer modes, the CNTRL1 input must be high when the command is issued. The command causes the following:

1. VSYNC and HSYNC are driven low for the duration of RESET and BLANK goes

high. BLANK remains high until a 'display on' command is received.

2. The interrupt and status bits and masks are set to zero, except for the RDFLG flag which is set to a one.
3. The transparent mode, cursor off, display off, and light pen disable states are set.
4. The initialization register pointer is set to address IR0.

Load IR Address
This command is used to preset the initialization register pointer with the value 'V' defined by D3 – D0. Allowable values are 0 to 10.

Enable Light Pen
After invoking this command, receipt of a light pen strobe input will cause the light pen register to be loaded with the current buffer memory address and the corresponding interrupt and status flag to be set. Once loaded, further loads are inhibited until either one of the light pen registers are read or a reset function is performed.

Disable Light Pen
Light pen hits will not be recognized.

Display Off
Asserts the BLANK output. The DADD0 through DADD13 display address bus outputs may be optionally placed in the 3-State condition by setting bit 2 to a '1' when invoking the command.

Display On
Restores normal blanking operation either at the beginning of the next field (bit 2 = 1) or at the beginning of the next scan line (bit 2 = 0). Also returns the DADD0-DADD13 drivers to their active state.

Cursor Off
Disables cursor operation. Cursor output is placed in the low state.

Cursor On
Enables normal cursor operation.

Reset Interrupt/Status Bits
This command resets the designated bits in the interrupt and status registers. The bit positions correspond to the bit positions in the registers:

Bit 0 – Light pen

Bit 1 – Ready

Bit 2 – Split screen

Bit 3 – Line zero

Bit 4 – Vertical blank

Disable Interrupts
Sets the interrupt mask to zeros for the designated conditions, thus disabling these conditions from asserting the $\overline{\text{INTR}}$ output. Bit position correspondence is as above.

Enable Interrupts
Resets the selected interrupt and status register bits and writes the associated interrupt mask bits to a one. This enables the corresponding conditions to assert the $\overline{\text{INTR}}$ output. Bit position correspondence is as above.

Delayed Commands

This group of commands is utilized for the independent buffer mode of operation, although the 'increment cursor' command can also be used in other modes. With the exception of the 'write from cursor to pointer' and 'increment cursor' commands, all the commands of this type will be executed immediately or will be delayed depending on when the command is invoked. If invoked during the active screen time, the command is executed at the next horizontal blanking interval. If invoked during a vertical retrace interval or a 'display off' state, the command is executed immediately.

The 'increment cursor' and 'write from cursor to pointer' commands are executed immediately after they are issued. 'Increment cursor' requires approximately three $\overline{\text{CCLK}}$ periods for completion. 'Write from cursor to pointer' asserts the BLANK output during its execution. BLANK will not be released until the beginning of the horizontal blanking interval following the last write operation. This will allow more than one 'write from cursor to pointer' command to be executed during one frame and will blank the screen for the time required to execute the command.

In all cases, the PVTC will assert the READY/RDFLG status to signify completion of the command. No other commands should be given until the current command is completed. Therefore, the READY interrupt or RDYFLG status flag should be used for handshaking control between the PVTC and CPU when using these commands.

Read/Write at Pointer
Transfers data between the display buffer the bus interface latch using the address contained in the pointer register.

Table 6. PVTC Command Formats

D7	D6	D5	D4	D3	D2	D1	D0	Hex	COMMAND
Instantaneous Commands:									
0	0	0	0	0	0	0	0		Master reset
0	0	0	1	V	V	V	V		Load IR pointer with value V (V = 0 to 10)
0	0	1	d	d	d	1	0[1]		Disable light pen
0	0	1	d	d	d	1	1[2]		Enable light pen
0	0	1	d	1	N	d	0[1]		Display off. Float DADD bus if N = 1
0	0	1	d	1	N	d	1[2]		Display on: Next field (N = 1) or scan line (N = 0)
0	0	1	1	d	d	d	0[1]		Cursor off
0	0	1	1	d	d	d	1[2]		Cursor on
0	1	0	N	N	N	N	N		Reset interrupt/status: Bit reset where N = 1
1	0	0	N	N	N	N	N		Disable interrupt: Disable where N = 1
0	1	1	N	N	N	N	N		Enable interrupt: Enables interrupts and resets the corresponding interrupt/status bits where N = 1
			V	L	S	R	L		
			B	Z	S	D	P		
Delayed Commands:								**Hex**	
1	0	1	0	0	1	0	0	A4	Read at pointer address
1	0	1	0	0	0	1	0	A2	Write at pointer address
1	0	1	0	1	0	0	1	A9	Increment cursor address
1	0	1	0	1	1	0	0	AC	Read at cursor address
1	0	1	0	1	0	1	0	AA	Write at cursor address
1	0	1	0	1	1	0	1	AD	Read at cursor address and increment address
1	0	1	0	1	0	1	1	AB	Write at cursor address and increment address
1	0	1	1	1	0	1	1	BB	Write from cursor address to pointer address

NOTES:
1. Any combination of these three commands is valid.
2. Any combination of these three commands is valid.
3. d = Don't care.

Programmable Video Timing Controller (PVTC) SCN2672

Read/Write at Cursor
Transfers data between the display buffer and the bus interface latch using the address contained in the cursor register.

Increment Cursor
Adds one (modulo 16k) to the cursor address register.

Read/Write at Cursor and Increment
Transfers data between the display buffer and the bus interface latch using the address contained in the cursor register and then adds one (modulo 16k) to the cursor address register.

Write from Cursor to Pointer
Writes the data contained in the bus interface latch into the block of display memory designated by the cursor address and pointer address registers, inclusive. After completion of the command, the pointer address will be unchanged, but the cursor register contents will be equal to the pointer address.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +6	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5.0V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		0.2			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except $\overline{INTR}$ output)	$I_{OH} = -200\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-State leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{OD}	$\overline{INTR}$ open drain output leakage current	$V_O = 0$ to V_{CC}			10	μA
I_{CC}	Power supply current				160	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Rating may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND).
6. Typical values are at +25°C, typical processing parameters.
7. For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2V and output voltages of 0.8V and 2V as appropriate.
8. Test condition for outputs: $C_L = 150pF$.
9. Timing is illustrated and specified to $\overline{WR}$ and $\overline{RD}$ inputs. Device may also be operated with $\overline{CE}$ as the 'strobing' input. In this case, all timing specifications apply referenced to falling and rising edges of $\overline{CE}$.
10. This specification requires that the $\overline{CE}$ input be negated (high) between read and/or write cycles.
11. $\overline{BCE}$, $\overline{WDB}$,and $\overline{RDB}$ delays track each other within 10ns. Also, these output delays will tend to follow direction (min/max) of DADD0 – 13 delays.
12. These values were measured with a capacitance load of 150pF. To adjust the output delay,use the following correction factor: $50pF \leqslant C_L < 150pF$: −0.15ns/pF.

Programmable Video Timing Controller (PVTC) SCN2672

AC ELECTRICAL CHARACTERISTICS $T_A = 0$ to $+70°C$, $V_{CC} = +5.0V \pm 5\%$[4, 5, 6, 7, 8]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS				UNIT
			2.7MHz		4.0MHz		
			Min	Max	Min	Max	
Bus timing (Figure 15)[9]							
t_{AS}	A0 – A2 setup time to $\overline{\text{WR}}$,$\overline{\text{RD}}$ low		30		30		ns
t_{AH}	A0 – A2 hold time from $\overline{\text{WR}}$,$\overline{\text{RD}}$ high		0		0		ns
t_{CS}	$\overline{\text{CE}}$ setup time to $\overline{\text{WR}}$,$\overline{\text{RD}}$ low		0		0		ns
t_{CH}	$\overline{\text{CE}}$ hold time from $\overline{\text{WR}}$,$\overline{\text{RD}}$ high		0		0		ns
t_{RW}	$\overline{\text{WR}}$,$\overline{\text{RD}}$ pulse width		250		250		ns
t_{DD}	Data valid after $\overline{\text{RD}}$ low			200		200	ns
t_{DF}	Data bus floating after $\overline{\text{RD}}$ high			100		100	ns
t_{DS}	Data setup time to $\overline{\text{WR}}$ high		150		150		ns
t_{DH}	Data hold time from $\overline{\text{WR}}$ high		10		5		ns
t_{CC}	High time from $\overline{\text{CE}}$ to $\overline{\text{CE}}$[10]						
	Consecutive commands		600		600		ns
	Other accesses		300		300		ns
$\overline{\text{CCLK}}$ timing (Figures 16 and 17)							
t_{CCP}	$\overline{\text{CCLK}}$ period		370		250		ns
t_{CCH}	$\overline{\text{CCLK}}$ high time		125		100		ns
t_{CCL}	$\overline{\text{CCLK}}$ low time		125		100		ns
	Output delay from $\overline{\text{CCLK}}$ edge[12]						ns
t_{CCD1}	DADD0-13,MBC		40	175	40	150	ns
t_{CCD2}	BLANK,HSYNC,VSYNC/CSYNC, CURSOR, $\overline{\text{BEXT}}$,$\overline{\text{BREQ}}$,$\overline{\text{BACK}}$, $\overline{\text{BCE}}$,$\overline{\text{WDB}}$, $\overline{\text{RDB}}$[11]		40	225	40	200	ns
Other timings (Figures 17 and 18)							
t_{RDL}	READY/RDFLG low from $\overline{\text{WR}}$ high[9]			t_{CCP} + 30		t_{CCP} + 30	ns
t_{BAK}	$\overline{\text{BACK}}$ high from $\overline{\text{PBREQ}}$ low			225		200	ns
t_{BEXT}	$\overline{\text{BEXT}}$ high from $\overline{\text{PBREQ}}$ high			225		200	ns
t_{LPS}	Light pen strobe setup time to $\overline{\text{CCLK}}$ low		120		120		ns
t_{LPH}	Light pen strobe hold time from $\overline{\text{CCLK}}$ low		– 10		– 10		ns
t_{IRL}	$\overline{\text{INTR}}$ low from $\overline{\text{CCLK}}$ low			225		200	ns
t_{IRH}	$\overline{\text{INTR}}$ high from $\overline{\text{WR}}$,$\overline{\text{RD}}$ high[9]			600		600	ns

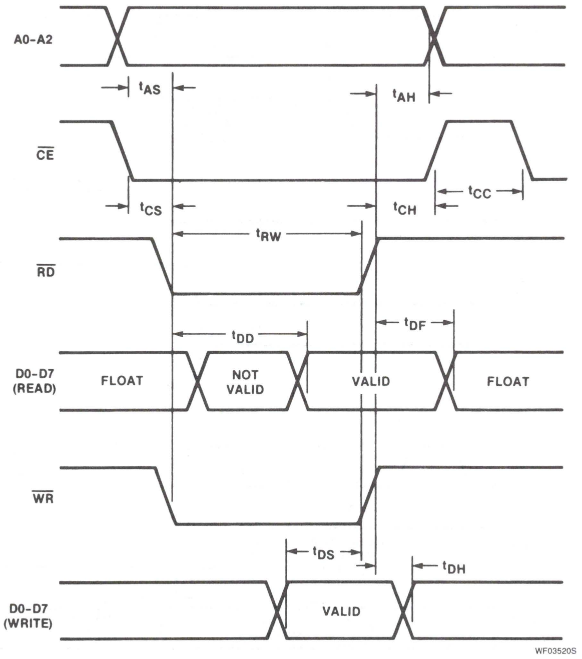
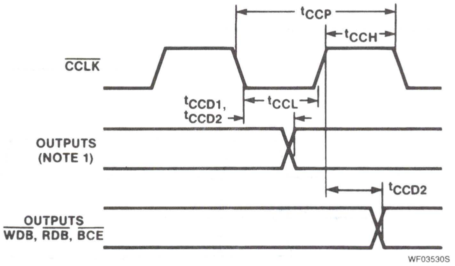

A0-A2
t_{AS}
t_{AH}
$\overline{CE}$
t_{CS}
t_{CH}
t_{CC}
t_{RW}
$\overline{RD}$
t_{DD}
t_{DF}
D0-D7 (READ)
FLOAT
NOT VALID
VALID
FLOAT
$\overline{WR}$
t_{DS}
t_{DH}
D0-D7 (WRITE)
VALID
WF03520S

Figure 15. Bus Timing


t_{CCP}
t_{CCH}
$\overline{CCLK}$
t_{CCD1},
t_{CCD2}
t_{CCL}
OUTPUTS (NOTE 1)
t_{CCD2}
$\overline{\text{OUTPUTS}}$ $\overline{WDB}$, $\overline{RDB}$, $\overline{BCE}$
WF03530S

NOTES:
1. DADD0 – DADD13, BLANK, HSYNC, CSYNC/VSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BCE}$, MBC, $\overline{BACK}$.
2. $\overline{BCE}$ changes state on both $\overline{CCLK}$ edges — (see Figures 3 and 4).

Figure 16. $\overline{CCLK}$ Timing

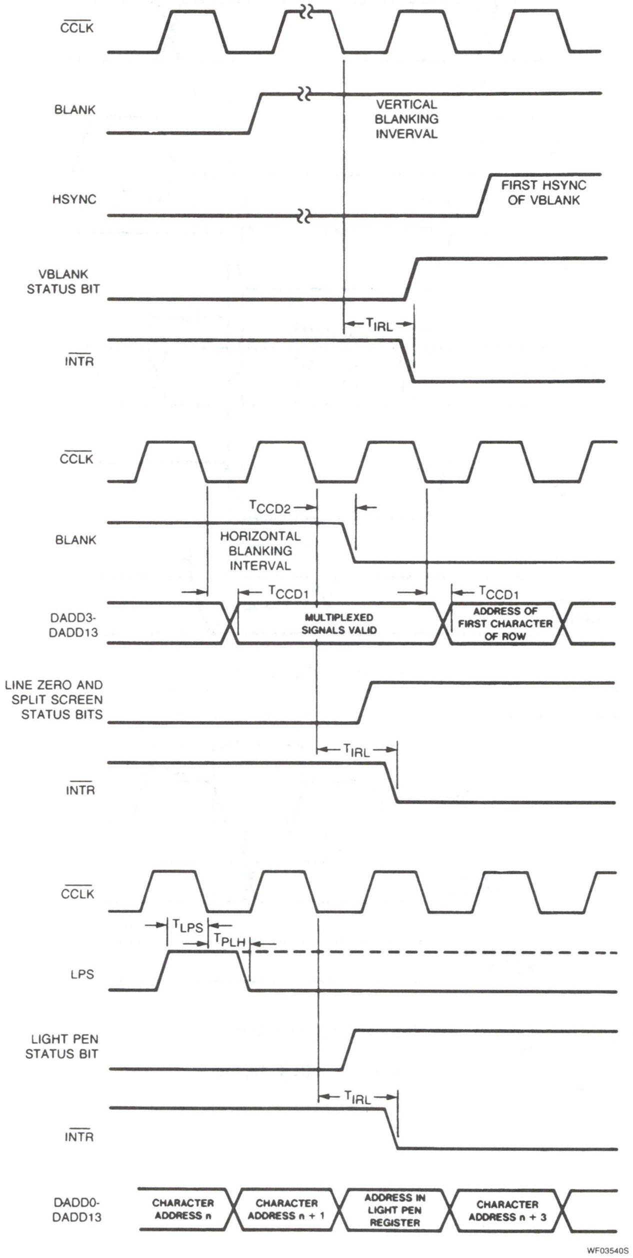

CCLK
BLANK
VERTICAL BLANKING INVERVAL
HSYNC
FIRST HSYNC OF VBLANK
VBLANK STATUS BIT
T_{IRL}
INTR
CCLK
T_{CCD2}
BLANK
HORIZONTAL BLANKING INTERVAL
T_{CCD1}
DADD3-DADD13
MULTIPLEXED SIGNALS VALID
T_{CCD1}
ADDRESS OF FIRST CHARACTER OF ROW
LINE ZERO AND SPLIT SCREEN STATUS BITS
T_{IRL}
INTR
CCLK
T_{LPS}
T_{PLH}
LPS
LIGHT PEN STATUS BIT
T_{IRL}
INTR
DADD0-DADD13
CHARACTER ADDRESS n
CHARACTER ADDRESS n + 1
ADDRESS IN LIGHT PEN REGISTER
CHARACTER ADDRESS n + 3
WF03540S

Figure 17. Other Timings

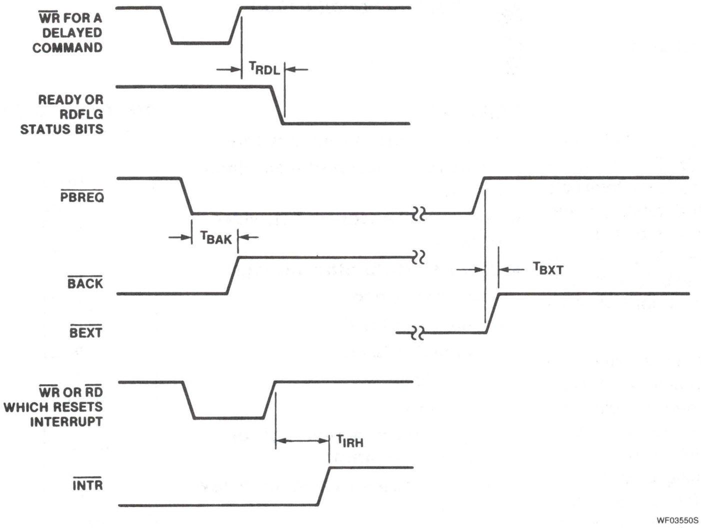
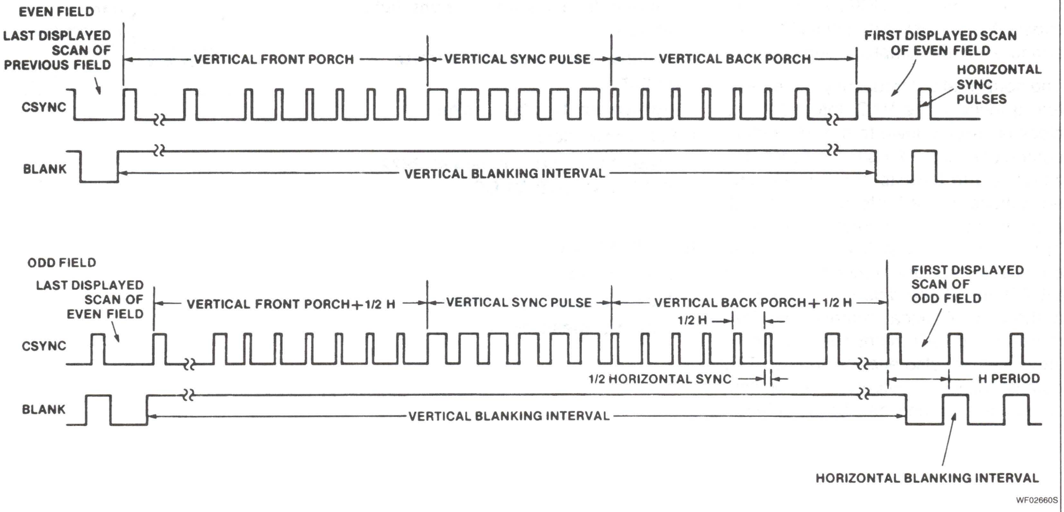

WR FOR A DELAYED COMMAND
TRDL
READY OR RDFLG STATUS BITS
PBREQ
TBAK
BACK
TBXT
BEXT
WR OR RD WHICH RESETS INTERRUPT
TIRH
INTR
WF03550S

Figure 18. Other Timings


EVEN FIELD
LAST DISPLAYED SCAN OF PREVIOUS FIELD
VERTICAL FRONT PORCH
VERTICAL SYNC PULSE
VERTICAL BACK PORCH
FIRST DISPLAYED SCAN OF EVEN FIELD
HORIZONTAL SYNC PULSES
CSYNC
BLANK
VERTICAL BLANKING INTERVAL
ODD FIELD
LAST DISPLAYED SCAN OF EVEN FIELD
VERTICAL FRONT PORCH+1/2 H
VERTICAL SYNC PULSE
VERTICAL BACK PORCH+1/2 H
1/2 H
FIRST DISPLAYED SCAN OF ODD FIELD
CSYNC
1/2 HORIZONTAL SYNC
H PERIOD
BLANK
VERTICAL BLANKING INTERVAL
HORIZONTAL BLANKING INTERVAL
WF02660S

NOTES:

1. In non-interlaced operation the even field is repeated continuously, and the odd field is not.
2. In interlaced operation the even field alternates with the odd field.

Figure 19. Composite Sync Timing

SCB2673
Video Attributes Controller (VAC)

Product Specification

DESCRIPTION

The Signetics 2673A and 2673B Video Attributes Controllers (VAC) are bipolar LSI devices designed for CRT terminals and display systems that employ raster scan techniques. Each contains a high-speed video shift register, field and character attributes logic, attribute latch, cursor format logic and half-dot shift control.

The VAC provides control of visual attributes on a field or character-by-character. Internal logic preserves field attribute data from character row to character row so that an attribute byte is not required at the beginning of each row. The 2673B provides for reverse video, blank (non-display), blink, underline and highlight attributes and a graphics mode attribute to work in conjunction with the Signetics 2670 Display Character and Graphics Generator (DCGG). The 2673A substitutes a light pen (strike-through) attribute for the graphics attribute.

The horizontal dot frequency is the basic timing input to the VAC. Internally, this clock is divided down to provide a character clock output for system synchronization. Up to ten bits of video dot data are parallel loaded into the video shift register on each character boundary. The video data is shifted out on three outputs at the dot frequency. On the VIDEO output, the data is presented as a three level signal representing low, medium and high intensities. The three intensities are also encoded on two TTL compatible video outputs. Light or dark screen background can be selected.

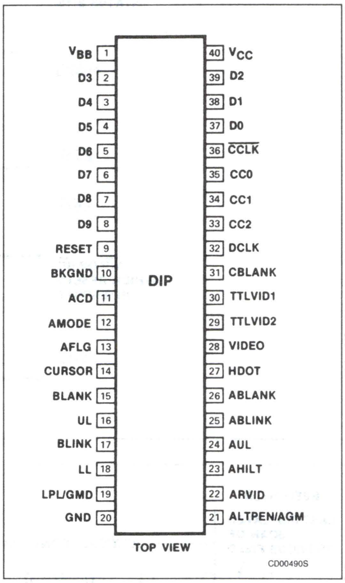
FEATURES

- **18 and 25MHz video dot rates**
- **Three-level current-driven video output**
- **Three level-encoded TTL video outputs**
- **Character/field attribute logic:**
  - **Reverse video**
  - **Character blank**
  - **Character blink**
  - **Underline**
  - **Highlight**
  - **Light pen strike-through or graphics control**
- **Field attributes extend from row to row**
- **Light or dark field**
- **Cursor reverse video logic**
- **Up to 10 dots per character**
- **Composite blanking for light field retrace**
- **Optional field graphics control output**
- **High-speed bipolar design**
- **TTL compatible**
- **Compatible with Signetics' 2672 PVTC and 2670 DCGG**

APPLICATIONS

- **CRT terminals**
- **Word processing systems**
- **Small business computers**

PIN CONFIGURATION

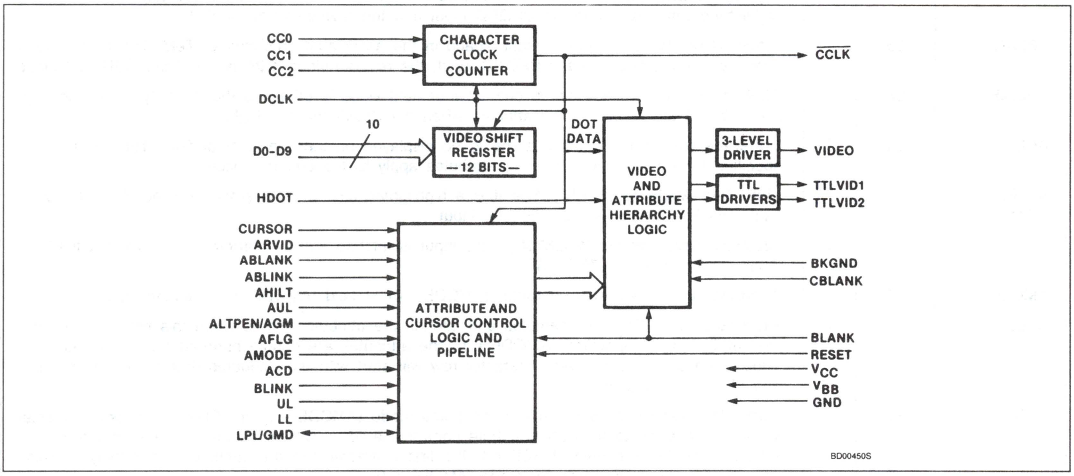

VBB 1
D3 2
D4 3
D5 4
D6 5
D7 6
D8 7
D9 8
RESET 9
BKGND 10
ACD 11
AMODE 12
AFLG 13
CURSOR 14
BLANK 15
UL 16
BLINK 17
LL 18
LPL/GMD 19
GND 20
40 VCC
39 D2
38 D1
37 D0
36 CCLK
35 CC0
34 CC1
33 CC2
32 DCLK
31 CBLANK
30 TTLVID1
29 TTLVID2
28 VIDEO
27 HDOT
26 ABLANK
25 ABLINK
24 AUL
23 AHILT
22 ARVID
21 ALTPEN/AGM
DIP
TOP VIEW
CD00490S

Video Attributes Controller (VAC) SCB2673

ORDERING INFORMATION

PACKAGES	$V_{CC} = +5V \pm 5\%$, $T_A = 0°C$ to $+70°C$			
	Graphics Attribute		Light Pen Attribute	
	25MHz	18MHz	25MHz	18MHz
Ceramic DIP	SCB2673BC5I40	SCB2673BC8I40	SCB2673AC5I40	SCB2673AC8I40
Plastic DIP	SCB2673BC5N40	SCB2673BC8N40	SCB2673AC5N40	SCB2673AC8N40

BLOCK DIAGRAM


CC0
CC1
CC2
CHARACTER CLOCK COUNTER
$\overline{\text{CCLK}}$
DCLK
10
D0-D9
VIDEO SHIFT REGISTER —12 BITS—
DOT DATA
VIDEO AND ATTRIBUTE HIERARCHY LOGIC
3-LEVEL DRIVER
VIDEO
TTL DRIVERS
TTLVID1
TTLVID2
HDOT
CURSOR
ARVID
ABLANK
ABLINK
AHILT
AUL
ALTPEN/AGM
AFLG
AMODE
ACD
BLINK
UL
LL
LPL/GMD
ATTRIBUTE AND CURSOR CONTROL LOGIC AND PIPELINE
BKGND
CBLANK
BLANK
RESET
V_{CC}
V_{BB}
GND
BD00450S

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
	DIP		
DCLK	32	I	**Dot Clock:** Dot frequency input. Video shift output rate.
$\overline{\text{CCLK}}$	36	O	**Character Clock:** A submultiple of DCLK. The frequency ranges from one sixth to one twelfth of DCLK, as determined by the state of the CC0 – CC2 inputs.
CC2 – CC0	33 – 35	I	**Character Clock Control:** The logic state on these three static inputs determine the internal divide factor for the $\overline{\text{CCLK}}$ output rate. Character clock rates of 6 through 12 dots per character may be specified.
D0 – D9	37 – 39, 2 – 8		**Dot Data Input:** These are parallel inputs corresponding to the character graphic symbol dot data for a given scan line. These inputs are strobed into the video shift register on the falling edge of each character clock.
HDOT	27	I	**Half-Dot Shift:** When this input is high, the serial video output is delayed by one half-dot time. This input is latched on the falling edge of each character clock.
CURSOR	14	I	**Cursor Timing:** This input provides the timing for the cursor video. When high, effectively reverses the intensities of the video and attributes. Cursor position, shape, and blink rate are controlled by this input.
BKGND	10	I	**Background Intensity:** Specifies light or dark video during BLANK and character fields. Affects the intensities of all attributes.
BLANK	15	I	**Screen Blank:** When high, this input forces the video ouputs to the level specified by the BKGND input (either high or low intensity). Not effective when CBLANK is high.

Video Attributes Controller (VAC) SCB2673

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
	DIP		
CBLANK	31	I	**Composite Blank:** Used with the TTL video outputs only. When high, this input forces the video outputs to a low intensity state for retrace blanking. When BKGND input is low, or when using video outputs, this input may be tied low.
ARVID	22	I	**Reverse Video Attribute:** The intensity of the associated character or field video is reversed. All other attributes are effectively reversed.
AHILT	23	I	**Highlight Attribute:** All dot video (including underline) of the associated character or field is highlighted with respect to the BKGND input and the reverse video attribute.
ABLANK	26	I	**Blank Attribute:** Generates a blank space in the associated character or field. The blank space intensity is determined by the BKGND input, the reverse video attribute, and the CURSOR input.
ABLINK	25	I	**Blink Attribute:** The associated character or field video is driven to the intensity determined by BKGND and the reverse video attribute when the BLINK input is high.
AUL	24	I	**Underline Attribute:** Specifies a line to be displayed on the character or field. The line is specified by the UL input. All other attributes apply to the underline video.
ALTPEN/ AGM	21	I	**Light Pen Attribute (2673A):** Specifies a highlighted line to be displayed on the character or field. The line is specified by the LPL input.
		I	**Attribute Graphics Mode (2673B):** This input is latched and synchronized to provide a field GMD output for the 2670 DCGG.
AMODE	12	I	**Attribute Mode:** Specifies character (AMODE = 0) or field (AMODE = 1) attributes mode.
AFLG	13	I	**Attributes Flag:** The VAC samples and latches the attributes inputs when this input is high. If field attributes are specified (AMODE = 1), the attributes are double buffered on a row basis. Thus, each scan line of every character row will start with the attributes that were valid at the end of the previous row.
ACD	11	I	**Attributes Control Display:** In field attributes mode (AMODE = 1), if ACD = 0, the first character in each new attribute field (the attribute control character) will be suppressed and only the attributes will be displayed. If ACD = 1, the first character and the attributes are displayed. This input has no effect in character mode (AMODE = 0).
BLINK	17	I	**Blink:** This input is sampled on the falling edge of BLANK to provide the blink rate for the character blink attribute. It should be a submultiple of the frame rate.
UL	16	I	**Underline:** Indicates the scan line(s) for the underline attribute. Latched on the falling edge of BLANK.
LPL/GMD	19	I	**Light Pen Line (2673A):** Indicates the scan line(s) for the light pen strike-through attribute. Latched on the falling edge of BLANK.
		O	**Graphics Mode (2673B):** This output provides a synchronized, latched, field graphics mode corresponding to the AGM input. This output can be used to control the GM input on the 2670 DCGG.
LL	18	I	**Last Line:** Indicates the last scan line of each character row. Used internally to extend field attributes across row boundaries. Latched on the falling edge of BLANK. This input has no effect in character mode (AMODE = 0).
VIDEO	28	O	**Video:** A three-level serial video output which corresponds to the composite dot pattern of characters, attributes and cursor.
TTLVID1	30	O	**TTL Video 1:** This output corresponds to the serial, non-highlighted video dot pattern.
TTLVID2	29	O	**TTL Video 2:** This output corresponds to the highlighted serial video dot pattern. Should be used with TTLVID1 to decode a composite video of three intensities.
RESET	9	I	**Manual Reset:** This active high input initializes the internal logic and resets the attribute latches.
V_{CC}	40	I	**Power Supply:** +5V ±5%
V_{BB}	1	I	**Bias Supply:** See Figure 13.
GND	20	I	**Ground:** 0V reference

Video Attributes Controller (VAC) SCB2673

FUNCTIONAL DESCRIPTION

The VAC consists of four major sections (see block diagram). The high-speed dot clock input is divided internally to provide a character clock for system timing. The parallel dot data is loaded into the video shift register on each character boundary and shifted into the video logic block at the dot rate. The six attribute inputs are latched internally and combined with the serial dot data to provide a three level video source for the monitor.

A separate BLANK input defines the active screen area. When BLANK = 0, the video levels are derived internally by the combinations of dot data, attributes, cursor, and the state of the BKGND input. Either black or white background can be selected. Symbols (dot data) are normally gray and can be highlighted to white or black as shown in Figure 1.

During the inactive screen area (BLANK = 1), the video level produced by the TTL outputs is either white (BKGND = 1) or black (BKGND = 0). A separate composite blank (CBLANK) input is provided to suppress raster retrace video when white background is specified. During the inactive screen area (BLANK = 1), the video level produced by the VIDEO output is either black (BKGND = 1) or white (BKGND = 0).

For the latter case, raster retrace video suppression is accomplished by raising the BKGND input during horizontal and vertical retrace intervals. For black background, tie BKGND high. Tie CBLANK input low for both cases.

Since BLANK is delayed by 3 CCLKs internally, CBLANK must be sync'ed with the internal BLANK signal to avoid active scan data from being suppressed. A CBLANK transition must occur at least 15ns before the rising edge of DCLK in order to be latched into the 2673 (see Figure 8).

Table 1. Clock Control Inputs

CC2	CC1	CC0	$\overline{\text{CCLK}}$ Dots/Character	$\overline{\text{CCLK}}$ Duty Cycle*
0	0	0	6	3/3
0	0	1	6	3/3
0	1	0	7	4/3
0	1	1	8	4/4
1	0	0	9	5/4
1	0	1	10	5/5
1	1	0	11	6/5
1	1	1	12	6/6

NOTE:
* High/low

Character Clock Counter

The character clock counter divides the frequency on the DCLK input to generate the character clock ($\overline{\text{CCLK}}$). The divide factor is specified by the clock control inputs (CC0 – CC2) as shown in Table 1.

Video Shift Register

On each character boundary, the parallel data (D0 – D9) is loaded into the video shift register. The data is shifted out least significant bit first (D0) by the DCLK. If 11 or 12 dots/character are specified (CC2 – CC0 = 110 or 111), a 0 (blank dot) is always shifted out before D0. For 12 dots/character, a 0 is also shifted out after D9. The serial dot data is shifted into the video logic where it is combined with the cursor and attributes to encode three levels of video.

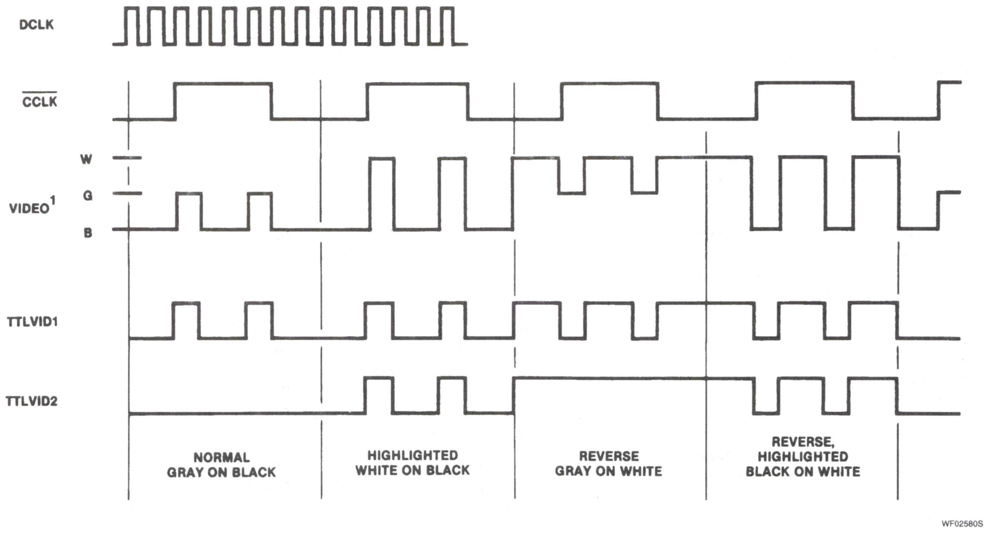
Attribute and Cursor Control

The VAC visual attributes capabilities include: reverse video, character blank, blink, underline, highlight, and light pen strike-through. The six attributes and the three attribute control inputs (AMODE, AFLG, and ACD) are clocked into the VAC on the falling edge of $\overline{\text{CCLK}}$. If AFLG is high, the attributes are latched internally and are effective for either one character time (AMODE = 0) or until another set of attributes is latched (AMODE = 1). The attributes set is double buffered on a row-by-row basis internally. Using this technique, field attributes can extend across character row boundaries, thereby eliminating the necessity of starting each row with an attribute set.

When field attribute mode is selected, (AMODE = 1), the VAC will accommodate two attribute storage configurations. In one configuration, the attribute control data is stored in the refresh RAM, taking the place of the first character code in the field to be affected. For this mode, the ACD input is tied low and blank characters will be displayed in the screen positions occupied by the attribute data (see Figure 11). In the second configuration, (ACD = 1), the character codes and attribute data are presented to the VAC in parallel. In this mode, dot data is displayed at each character position (see Figure 12).

The CURSOR and the attribute input signals are pipelined internally to allow for system propagations (one $\overline{\text{CCLK}}$ for refresh RAM, one $\overline{\text{CCLK}}$ for dot generator). The attribute timing signals BLINK, UL, LPL and LL are clocked into the VAC at the beginning of each scan line by the falling edge of the BLANK input. Thus, these signals must be in their proper state at the falling edge of BLANK preceding the scan line at which they are to be active (see Figure 4).

Video Attributes Controller (VAC) SCB2673



NOTES:
W = White
G = Gray
B = Black

Figure 1. Encoded Video Outputs

Video Logic

The serial dot data and the pipelined cursor and attributes are combined to generate the three level current source on the VIDEO output. The three levels (white, gray, and black) are also encoded on the two TTL compatible outputs TTLVID1 and TTLVID2. The three levels are encoded as shown in Table 2.

The video is normally shifted out on the leading edge of the DCLK. When the HDOT input is asserted, the corresponding dot data is delayed by one-half DCLK. This half-dot shifting, when used on selected lines of character video, can be used to effect character rounding as shown in Figure 2.

Attribute Hierarchy

The video of each character block consists of four components as shown in Figure 3.

Table 2. Video Output

TTLVID2	TTLVID1	INTENSITY
0	0	Black (or CBLANK)
0	1	Gray (on black surround)
1	0	Gray (on white surround)
1	1	White

NOTE:
The TTLVID1 output can be used independently to generate a two-level non-highlighted video.

Symbol video is generated from the dot data inputs D0 – D9.

Underline video is enabled by the AUL attribute and is generated when the UL timing input is active. Underline and symbol video are always the same intensity.

Strike-through video is enabled by the ALTPEN attribute and is generated when the LPL timing input is active. This video is always highlighted and takes precedence over the symbol and underline video. This feature applies to the 2673A only.

Surround video is the absence of symbol, underline and strike-through video or the presence of the non-display attributes (ABLANK or ABLINK • BLINK).

The relative intensities of the four video components are determined by the remaining attributes (AHILT, ABLANK, ABLINK, ARVID) and the BKGND and CURSOR inputs as illustrated in Table 3.

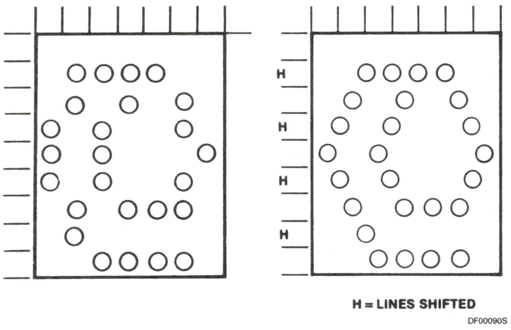

H
H
H
H
H = LINES SHIFTED
DF00090S

Figure 2. "AT" Symbol With and Without Half-Dot Shifting

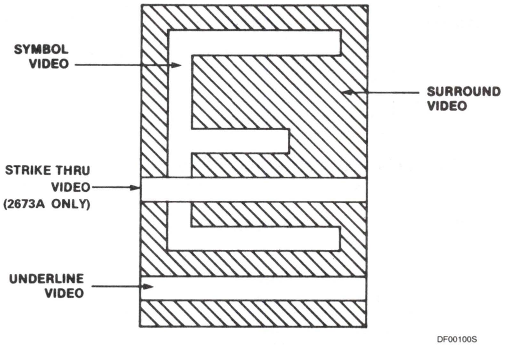

SYMBOL VIDEO
SURROUND VIDEO
STRIKE THRU VIDEO (2673A ONLY)
UNDERLINE VIDEO
DF00100S

Figure 3. Video Components of Character Block

Table 3. Attributes Hierarchy

ATTRIBUTES AND CONTROL INPUTS d = Don't Care				RELATIVE VIDEO INTENSITIES W = White, B = Black, G = Gray		
BKGND[5]	Reverse[1]	Non-Display[2]	AHILT	Strike-Through Video[3]	Symbol Or Underline Video[3, 4]	Surround Video[3]
0	0	0	0	W	G	B
0	0	0	1	W	W	B
0	0	1	d	B	B	B
0	1	0	0	B	G	W
0	1	0	1	B	B	W
0	1	1	d	W	W	W
1	0	0	0	B	G	W
1	0	0	1	B	B	W
1	0	1	d	W	W	W
1	1	0	0	W	G	B
1	1	0	1	W	W	B
1	1	1	d	B	B	B

NOTES:
1. Reverse = ARVID • $\overline{\text{CURSOR}}$ + $\overline{\text{ARVID}}$ • CURSOR
2. Non-display = ABLANK + ABLINK • BLINK
3. See Figure 3.
4. Symbol and underline video are always the same intensity.
5. Reverse sense for VIDEO output.

Video Attributes Controller (VAC) SCB2673

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$, V_{BB} = see Figure 14[3, 4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2			V
V_{OL}	Output low voltage (except VIDEO)	$I_{OL} = 4mA$			0.4	V
V_{OH}	Output high voltage (except VIDEO)	$I_{OH} = -400\mu A$	2.4			V
V_B	VIDEO black level	$R_L = 150\Omega$ to GND		0		V
V_G	VIDEO gray level	$R_L = 150\Omega$ to GND		0.45		V
V_W	VIDEO white level	$R_L = 150\Omega$ to GND		0.90		V
I_{IL}	Input low current	$V_{IN} = 0.4V$			−400/ −800[6]	μA
I_{IH}	Input high current	$V_{IN} = 2.4V$			20/40[6]	μA
I_{CC}	V_{CC} supply current	$V_{IN} = 0V$, V_{CC} = Max			80	mA
I_{BB}	V_{BB} supply current	V_{BB} = Max			120	mA

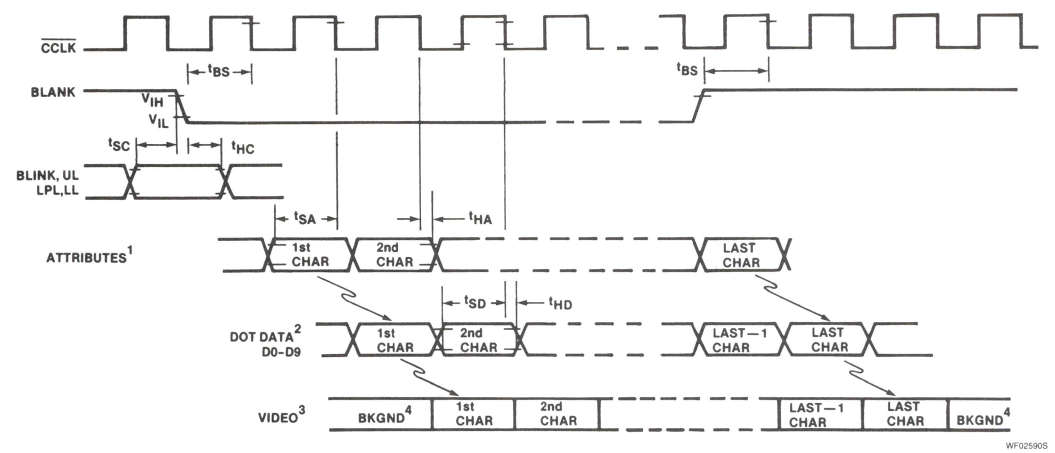
NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. Parameters are valid over operating temperature range unless otherwise specified.
4. All voltage measurements are referenced to ground (V_{SS}). All input signals swing between 0.4V and 2.4V. All time measurements are referenced at input voltages of 0.8V, 2V and at output voltages of 0.8V, 2V as appropriate.
5. Typical values are at +25°C, typical supply voltages and typical processing parameters.
6. For DCLK input.
7. C_L less than 150pF minimum could be faster.

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$, V_{BB} = see Figure 14[3, 4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS				UNIT
			25MHz Version		18MHz Version		
			Min	Max	Min	Max	
Dot clock (see Figure 10)							
f_d	Frequency (HDOT = 0)			25		18	MHz
	(HDOT = 1)			18		18	MHz
t_{DH}	High		15		22		ns
t_{DL}	Low		15		22		ns
Setup times to $\overline{\text{CCLK}}$ (see Figures 4, 5, 6 and 10)							
t_{BS}	BLANK		50		50		ns
t_{SC}	BLINK, UL, LPL, LL (ref to BLANK)		20		20		ns
t_{SA}	Attributes		45		55		ns
t_{SD}	Dot data D0 – D9		70		70		ns
t_{SK}	CURSOR		50		50		ns
t_{FS}	AFLG, AMODE		50		65		ns
t_{SH}	HDOT		45		55		ns
Hold times from $\overline{\text{CCLK}}$ (see Figures 4, 5, 6 and 10)							
t_{HC}	BLINK, UL, LPL, LL (ref to BLANK)		20		20		ns
t_{HA}	Attributes		20		20		ns
t_{HD}	Dot data D0 – D9		30		30		ns
t_{HK}	CURSOR		20		20		ns
t_{FH}	AFLG, AMODE		30		30		ns
t_{HH}	HDOT		20		20		ns
Setup times to DCLK (see Figure 9)							
t_{SG}	BKGND		15		15		ns
t_{SB}	CBLANK		15		15		ns
t_{CS}	CC0 – CC2		30		35		ns
Hold times from DCLK (see Figure 9)							
t_{HG}	BKGND		15		15		ns
t_{HB}	CBLANK		15		15		ns
t_{CH}	CC0 – CC2		20		20		ns
Delay times (see Figures 6 and 7)							
t_{DGM}	GMD from DCLK	$C_L = 150pF$		65		65	ns
t_{DC}	$\overline{\text{CCLK}}$ from DCLK			65		65	ns
t_{DV}[7]	TTLVID1 and TTLVID2 from DCLK			75		80	ns
t_{DV}	VIDEO from DCLK			240		240	ns

Video Attributes Controller (VAC) SCB2673


CCLK
BLANK
t_{BS}
V_{IH}
V_{IL}
t_{SC}
t_{HC}
BLINK, UL
LPL,LL
t_{SA}
t_{HA}
ATTRIBUTES^1
1st CHAR
2nd CHAR
LAST CHAR
t_{SD}
t_{HD}
DOT DATA^2
D0-D9
LAST−1 CHAR
VIDEO^3
BKGND^4
WF02590S

NOTES:
1. Attributes include: ABLINK, ABLANK, ARVID, AUL, AHILT, and ALTPEN.
2. One $\overline{CCLK}$ delay for dot data (obtained from delay through character generator).
3. See Figure 7 for detail timing of VIDEO, TTLVID1, TTLVID2.
4. Non-active scan time. VIDEO reverts to polarity selected by the BKGND input.

Figure 4. VAC Pipeline Timing

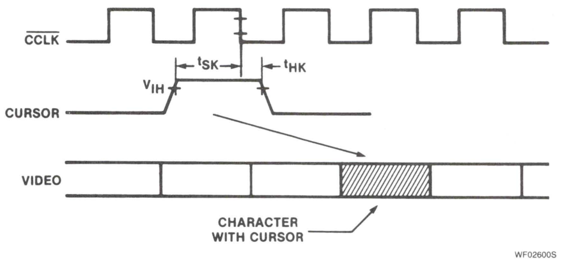
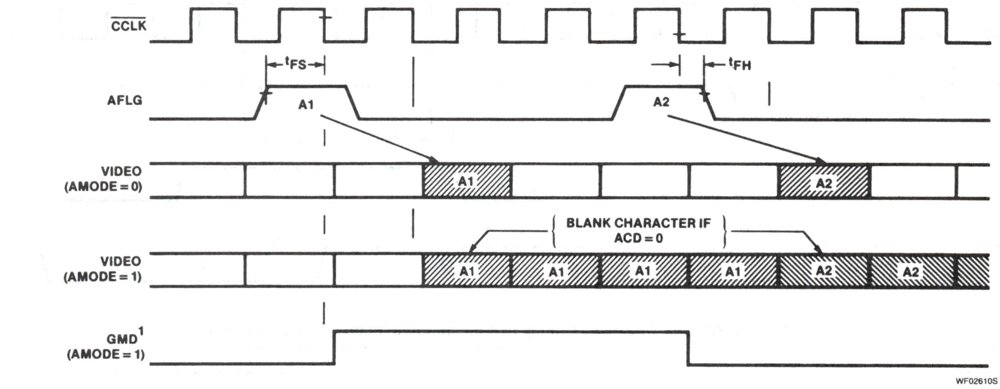
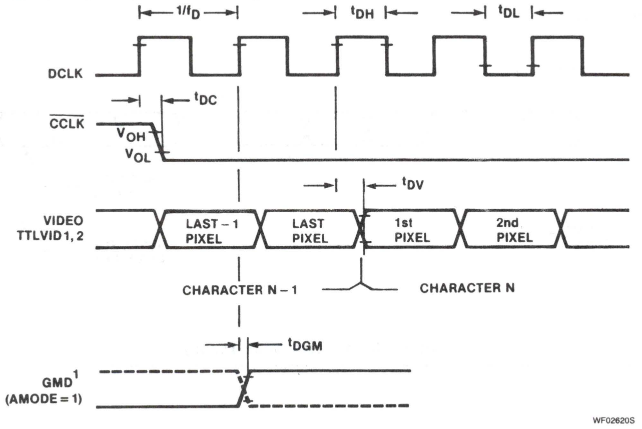

CCLK
t_{SK}
t_{HK}
V_{IH}
CURSOR
VIDEO
CHARACTER WITH CURSOR
WF02600S

Figure 5. Cursor Pipeline Timing


CCLK
tFS
tFH
AFLG
A1
A2
VIDEO (AMODE = 0)
BLANK CHARACTER IF ACD = 0
VIDEO (AMODE = 1)
GMD¹ (AMODE = 1)
WF02610S

NOTE:
1. GMD output in 2673B version only. See Figure 7 for detail timing.

Figure 6. Character (AMODE = 0), Field (AMODE = 1), and GMD Attribute Timing


1/fD
tDH
tDL
DCLK
tDC
CCLK
VOH
VOL
tDV
VIDEO TTLVID1, 2
LAST – 1 PIXEL
LAST PIXEL
1st PIXEL
2nd PIXEL
CHARACTER N – 1
CHARACTER N
tDGM
GMD¹ (AMODE = 1)
WF02620S

NOTE:
1. GMD output in 2673B version only.

Figure 7. Video and GMD Pipeline Timing

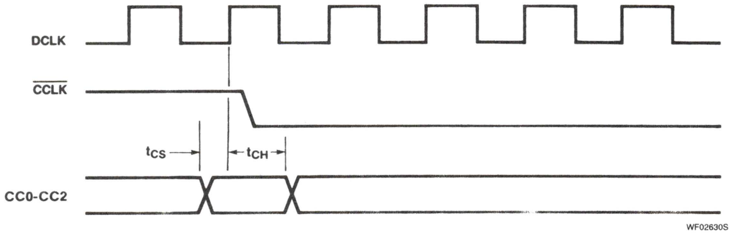

DCLK
CCLK
tCS
tCH
CC0-CC2
WF02630S

Figure 8. Character Clock Counter Timing

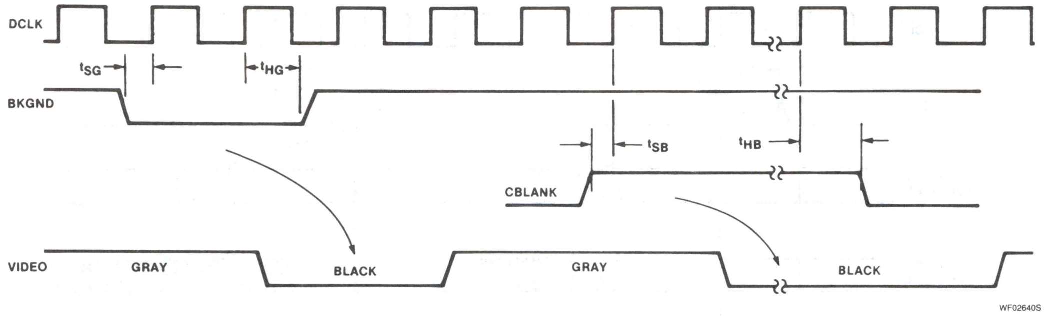
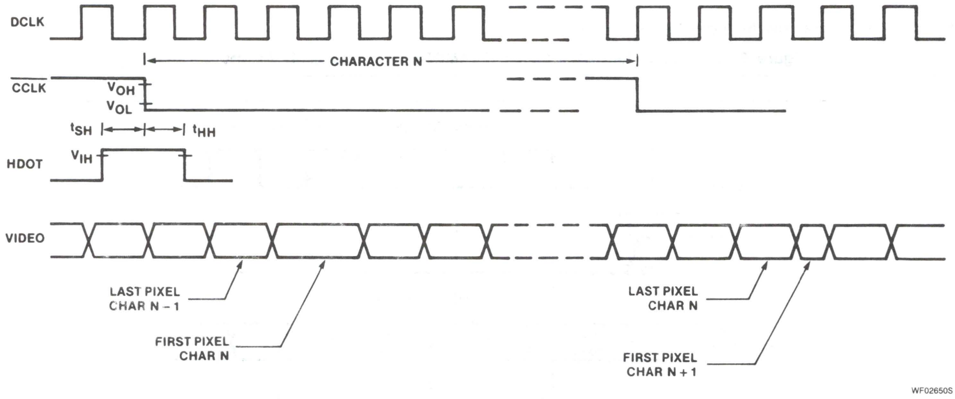

DCLK
t_{SG}
t_{HG}
BKGND
t_{SB}
t_{HB}
CBLANK
VIDEO
GRAY
BLACK
GRAY
BLACK
WF02640S

Figure 9. BKGND and CBLANK Timing During Inactive Scan Time (Blank = 1)


DCLK
CHARACTER N
CCLK
V_{OH}
V_{OL}
t_{SH}
t_{HH}
HDOT
V_{IH}
VIDEO
LAST PIXEL CHAR N − 1
FIRST PIXEL CHAR N
LAST PIXEL CHAR N
FIRST PIXEL CHAR N + 1
WF02650S

NOTE:
Half-dot shift feature 18MHz maximum.

Figure 10. Half-Dot Shift Timing

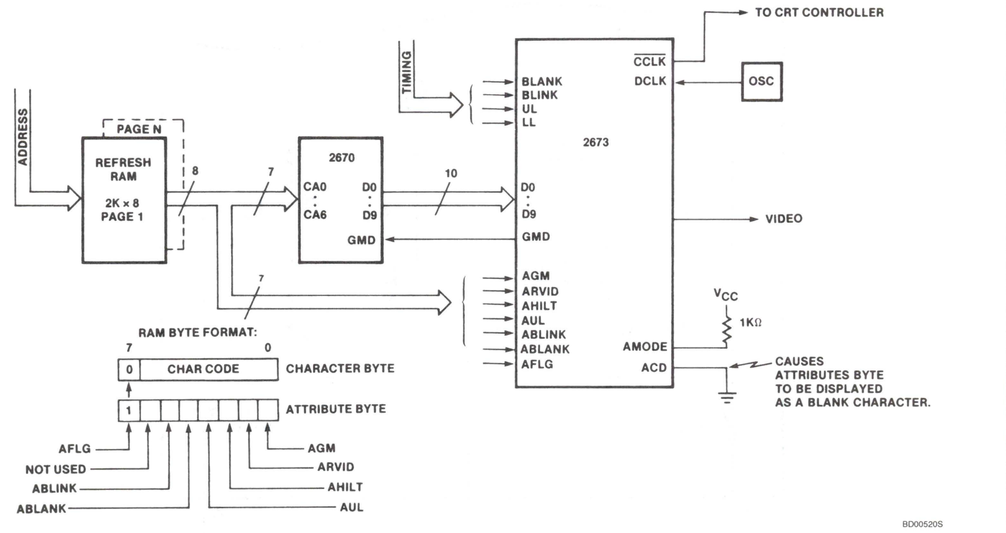
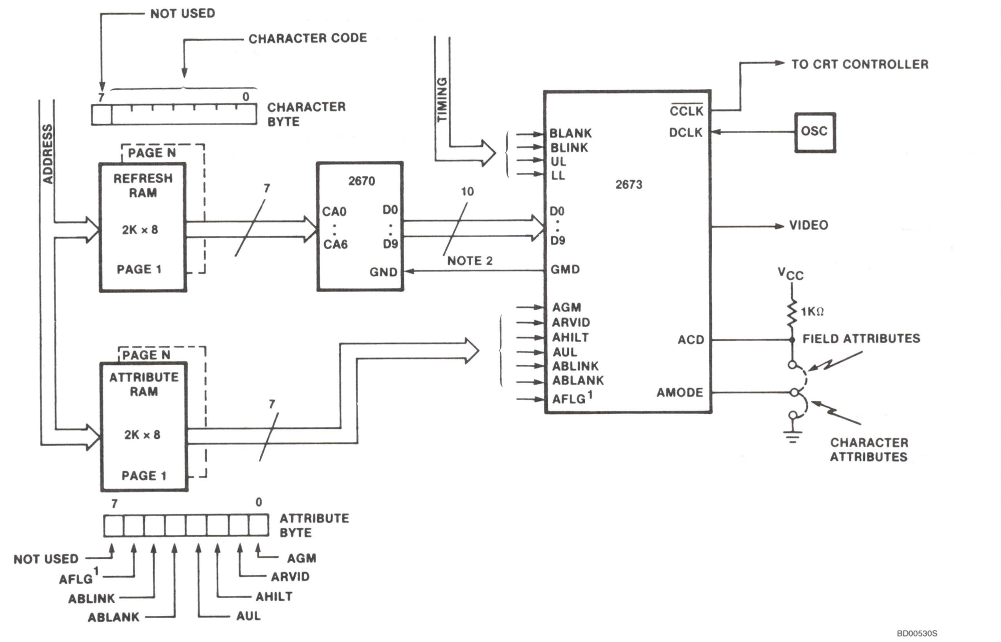

TO CRT CONTROLLER
TIMING
ADDRESS
PAGE N
REFRESH RAM
2K × 8
PAGE 1
8
7
2670
CA0
CA6
D0
D9
GMD
10
BLANK
BLINK
UL
LL
2673
CCLK
DCLK
OSC
D0
D9
GMD
VIDEO
7
AGM
ARVID
AHILT
AUL
ABLINK
ABLANK
AFLG
AMODE
ACD
VCC
1KΩ
CAUSES ATTRIBUTES BYTE TO BE DISPLAYED AS A BLANK CHARACTER.
RAM BYTE FORMAT:
7
0
0
CHAR CODE
CHARACTER BYTE
1
ATTRIBUTE BYTE
AFLG
NOT USED
ABLINK
ABLANK
AGM
ARVID
AHILT
AUL
BD00520S

Figure 11. System Block Diagram of the 2673 in Field Attribute Mode Using the Narrow RAM (8 Wide) Configuration


NOT USED
CHARACTER CODE
7
0
CHARACTER BYTE
TIMING
TO CRT CONTROLLER
ADDRESS
PAGE N
REFRESH RAM
2K × 8
PAGE 1
7
2670
CA0
CA6
D0
D9
GND
10
NOTE 2
BLANK
BLINK
UL
LL
2673
CCLK
DCLK
OSC
D0
D9
GMD
VIDEO
VCC
1KΩ
PAGE N
ATTRIBUTE RAM
2K × 8
PAGE 1
7
AGM
ARVID
AHILT
AUL
ABLINK
ABLANK
AFLG1
ACD
AMODE
FIELD ATTRIBUTES
CHARACTER ATTRIBUTES
7
0
ATTRIBUTE BYTE
NOT USED
AFLG1
ABLINK
ABLANK
AGM
ARVID
AHILT
AUL
BD00530S

NOTES:
1. For operation in character attribute mode, tie AFLG high.
2. In character attribute mode, AGM output from RAM should be connected directly to the 2670 GMD input.

Figure 12. System Block Diagram of the 2673 in Field or Character Attribute Mode Using the Wide RAM Configuration

Video Attributes Controller (VAC) SCB2673

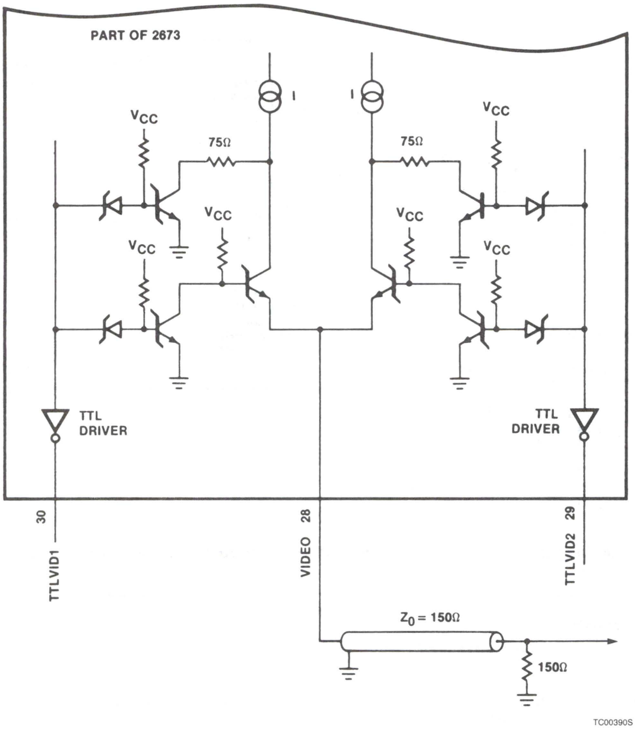

PART OF 2673
I
I
V_{CC}
V_{CC}
75Ω
75Ω
V_{CC}
V_{CC}
V_{CC}
V_{CC}
TTL DRIVER
TTL DRIVER
30
28
29
TTLVID1
VIDEO
TTLVID2
$Z_0 = 150Ω$
150Ω
TC00390S

Figure 13. Video Output Stages of the 2673

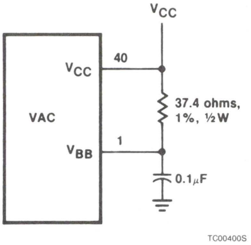

V_{CC}
40
V_{CC}
VAC
37.4 ohms, 1%, ½W
1
V_{BB}
0.1μF
TC00400S

Figure 14. Test Diagram

Signetics

SCN2674
Advanced Video Display Controller (AVDC)

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics SCN2674 Advanced Video Display Controller (AVDC) is a programmable device designed for use in CRT terminals and display systems that employ raster scan techniques. The AVDC generates the vertical and horizontal timing signals necessary for the display of interlaced or non-interlaced data on a CRT monitor. It provides consecutive addressing to a user-specified display buffer memory domain and controls the CPU display buffer interface for various buffer configuration modes. A variety of operating modes, display formats, and timing profiles can be implemented by programming the control registers in the AVDC.

A minimum CRT terminal system configuration consists of an AVDC, an SCN2671 Keyboard and Communication Controller (PKCC), an SCN2670 Display Character and Graphics Generator (DCGG), an SCB2675 Color/Monochrome Attributes Controller (CMAC), a single-chip microcomputer such as the 8048, a display buffer RAM, and a small amount of TTL for miscellaneous address decoding, interface, and control. Typically, the package count for a minimum system is between 15 and 20 devices; system complexity can be enhanced by upgrading the microprocessor and expanding via the system address and data buses.

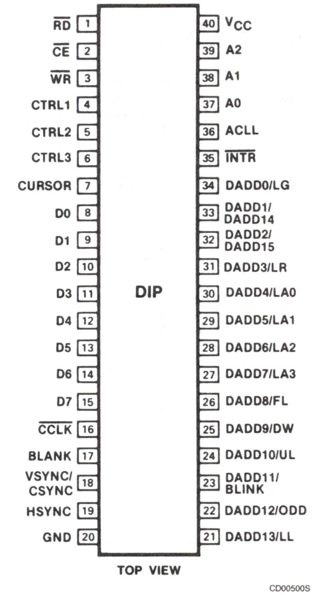
FEATURES

- **2.7MHz and 4MHz character rates**
- **1 to 256 characters per row**
- **1 to 16 raster lines per character row**
- **1 to 128 character rows per frame**
- **Bit-mapped graphics mode**
- **Programmable horizontal and vertical sync generators**
  - **RS-170 compatible sync**
- **Interlaced or non-interlaced operation**
- **Up to 64k RAM addressing for multiple page operation**
- **Readable, writable and incrementable cursor**
  - **Programmable cursor size and blink**
- **AC line lock**
- **Automatic wraparound of RAM**
- **Automatic split screen**
- **Automatic bidirectional soft scrolling**
  - **Programmable scan line increment**
- **Row table addressing mode**
- **Double height tops and bottoms**
- **Double width control output**
- **Selectable buffer interface modes**
- **Dynamic RAM refresh**
- **Completely TTL compatible**
- **Single +5V power supply**
- **Power-on reset circuit**

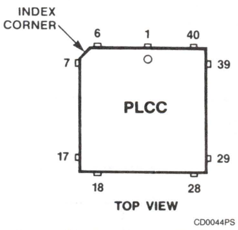
APPLICATIONS

- **CRT terminals**
- **Word processing systems**
- **Small business computers**
- **Home computers**

PIN CONFIGURATIONS



RD 1
CE 2
WR 3
CTRL1 4
CTRL2 5
CTRL3 6
CURSOR 7
D0 8
D1 9
D2 10
D3 11
D4 12
D5 13
D6 14
D7 15
CCLK 16
BLANK 17
VSYNC/ CSYNC 18
HSYNC 19
GND 20
40 VCC
39 A2
38 A1
37 A0
36 ACLL
35 INTR
34 DADD0/LG
33 DADD1/ DADD14
32 DADD2/ DADD15
31 DADD3/LR
30 DADD4/LA0
29 DADD5/LA1
28 DADD6/LA2
27 DADD7/LA3
26 DADD8/FL
25 DADD9/DW
24 DADD10/UL
23 DADD11/ BLINK
22 DADD12/ODD
21 DADD13/LL
DIP
TOP VIEW
CD00500S



INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

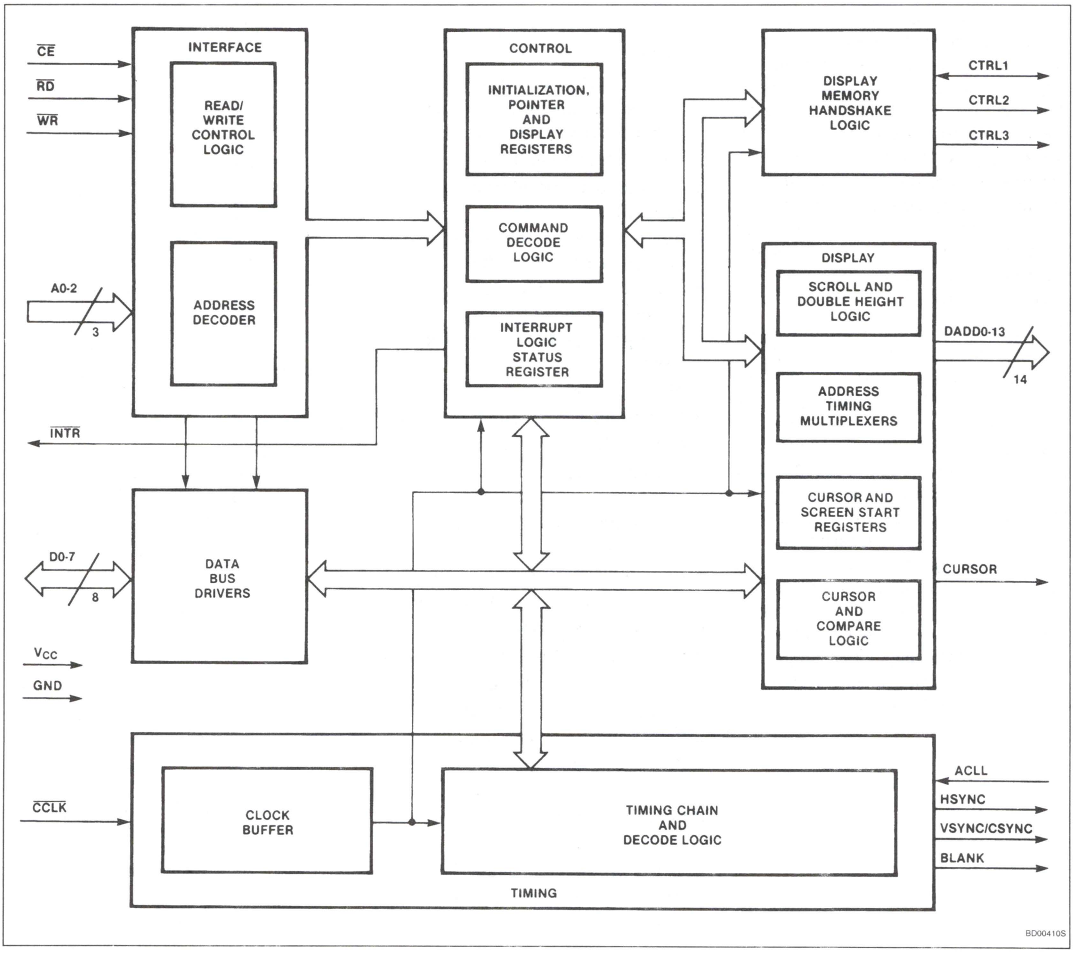
Pin	Function
1	NC
2	$\overline{RD}$
3	$\overline{CE}$
4	$\overline{WR}$
5	CTRL1
6	CTRL2
7	CTRL3
8	CURSOR
9	D0
10	D1
11	D2
12	NC
13	D3
14	D4
15	D5
16	D6
17	D7
18	$\overline{CCLK}$
19	BLANK
20	VSYNC CSYNC/
21	HSYNC
22	GND
23	NC
24	DADD13/LL
25	DADD12/ ODD
26	DADD11/ BLINK
27	DADD10/UL
28	DADD9/DW
29	DADD8/FL
30	DADD7/LA3
31	DADD6/LA2
32	DADD5/LA1
33	DADD4/LA0
34	NC
35	DADD3/LR
36	DADD2/ DADD15
37	DADD1/ DADD14
38	DADD0/LG
39	$\overline{INTR}$
40	ACLL
41	A0
42	A1
43	A2
44	V_{CC}

Advanced Video Display Controller (AVDC) SCN2674

ORDERING INFORMATION

PACKAGES	V_{CC} = +5V ±5%, T_A = 0°C to +70°C	
	4MHz	2.7MHz
Ceramic DIP	SCN2674BC4I40	SCN2674BC3I40
Plastic DIP	SCN2674BC4N40	SCN2674BC3N40
Plastic LCC	SCN2674BC4A44	SCN2674BC3A44

BLOCK DIAGRAM


CE
RD
WR
INTERFACE
READ/ WRITE CONTROL LOGIC
ADDRESS DECODER
A0-2
3
INTR
CONTROL
INITIALIZATION, POINTER AND DISPLAY REGISTERS
COMMAND DECODE LOGIC
INTERRUPT LOGIC STATUS REGISTER
DISPLAY MEMORY HANDSHAKE LOGIC
CTRL1
CTRL2
CTRL3
DISPLAY
SCROLL AND DOUBLE HEIGHT LOGIC
ADDRESS TIMING MULTIPLEXERS
CURSOR AND SCREEN START REGISTERS
CURSOR AND COMPARE LOGIC
DADD0-13
14
CURSOR
D0-7
8
DATA BUS DRIVERS
Vcc
GND
CCLK
CLOCK BUFFER
TIMING CHAIN AND DECODE LOGIC
TIMING
ACLL
HSYNC
VSYNC/CSYNC
BLANK
BD00410S

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
A0 – A2	37 – 39	41 – 43	I	**Address Lines:** Used to select AVDC internal registers for read/write operations and for commands.
D0 – D7	8 – 15	9 – 11, 13 – 17	I/O	**8-Bit Bidirectional Three-State Data Bus:** Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the AVDC take place over this bus. The direction of the transfer is controlled by the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ inputs when the $\overline{\text{CE}}$ input is low. When the $\overline{\text{CE}}$ input is high, the data bus is in the 3-State condition.
$\overline{\text{RD}}$	1	2	I	**Read Strobe:** Active low input. A low on this pin while $\overline{\text{CE}}$ is low causes the contents of the register selected by A0 – A2 to be placed on the data bus. The read cycle begins on the leading (falling) edge of $\overline{\text{RD}}$.
$\overline{\text{WR}}$	3	4	I	**Write Strobe:** Active low input. A low on this pin while $\overline{\text{CE}}$ is also low causes the contents of the data bus to be transferred to the register selected by A0 – A2. The transfer occurs on the trailing (rising) edge of $\overline{\text{WR}}$.
$\overline{\text{CE}}$	2	3	I	**Chip Enable:** Active low input. When low, data transfers between the CPU and the AVDC are enabled on D0 – D7 as controlled by the $\overline{\text{WR}}$, $\overline{\text{RD}}$, and A0 – A2 inputs. When $\overline{\text{CE}}$ is high, effectively, the AVDC is isolated from the data bus and D0 – D7 are placed in the 3-State condition.
$\overline{\text{CCLK}}$	16	18	I	**Character Clock:** Timing signal derived from the video dot clock which is used to synchronize the AVDC's timing functions.
HSYNC	19	21	O	**Horizontal Sync:** Active high output which provides video horizontal sync pulses. The timing parameters are programmable.
VSYNC/ CSYNC	18	20	O	**Vertical Sync/Composite Sync:** A control selects either vertical or composite sync pulses on this active high output. When CSYNC is selected, equalization pulses are included. The timing parameters are programmable.
BLANK	17	19	O	**Blank:** This active high output defines the horizontal and vertical borders of the display. Display control signals which are output on DADD0 through DADD13 are valid on the trailing edge of BLANK.
CURSOR	7	8	O	**Cursor Gate:** This output becomes active for a specified number of scan lines when the address contained in the cursor register matches the address output on DADD0 through DADD13 for displayable character addresses. The first and last lines of the cursor and a blink option are programmable. When the row table addressing mode is enabled, this output is active for a portion of the blanking interval prior to the first scan line of a character row, while the AVDC is fetching the starting address for that row.
$\overline{\text{INTR}}$	35	39	O	**Interrupt Request:** Open drain output which supplies an active low interrupt request from any of five maskable sources. This pin is inactive after a power on reset or a master reset command.
ACLL	36	40	I	**AC Line Lock:** If this input is low after the programmed vertical front porch interval, the vertical front porch will be lengthened by increments of horizontal scan line times until this input goes high. This input should be pulled high if not being used.
CTRL1	4	5	I/O	**Handshake Control 1:** In independent mode, provides an active low write data buffer ($\overline{\text{WDB}}$) output which strobes data from the interface latch into the display memory. In transparent and shared modes, this is an active low processor bus request ($\overline{\text{PBREQ}}$) input which indicates that the CPU desires to access the display memory.
CTRL2	5	6	O	**Handshake Control 2:** In independent mode, provides an active low read data buffer ($\overline{\text{RDB}}$) output which strobes data from the display memory into the interface latch. In transparent and shared modes, this is an active low bus external enable ($\overline{\text{BEXT}}$) output which indicates that the AVDC has relinquished control of the display memory (DADD0 – DADD13 are in the 3-State condition) in response to a CPU bus request. $\overline{\text{BEXT}}$ also goes low in response to a 'display off and float DADD' command. In row buffer mode, it is an active low bus request ($\overline{\text{BREQ}}$) output which halts the CPU during a line DMA.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
CTRL3	6	7	O	**Handshake Control 3:** In independent mode, provides the active low buffer chip enable ($\overline{\text{BCE}}$) signal to the display memory. In transparent and shared modes, provides an active low bus acknowledge ($\overline{\text{BACK}}$) output which serves as a ready signal to the CPU in response to a processor bus request. In row buffer mode, this is an active high memory bus control (MBC) output which configures the system for the DMA transfer of one row of character codes from system memory to the row display buffer.
DADD0 – DADD13	34-21	38 – 35, 33 – 24	O	**Display Address:** Used by the AVDC to address up to 16k of display memory directly, or up to 64k of memory by demultiplexing DADD14 and DADD15. These outputs are floated at various times depending on the buffer mode. Various control signals are multiplexed on DADD0 through DADD13 and are valid at the trailing edge of BLANK. These control signals are: **DADD0/LG** **Line Graphics:** Output which denotes bit-mapped graphic mode. **DADD1/DADD14** **Display Address 14:** Multiplexed address bit used to extend addressing to 64k. **DADD2/DADD15** **Display Address 15:** Multiplexed address bit used to extend addressing to 64k. **DADD3/LR** **Last Row:** Output which indicates the last active character row of each field. **DADD4 – DADD7/LA0 – LA3** **Line Address:** Provides the number of the current scan line count for each character row. **DADD8/FL** **First Line:** Asserted during the blanking interval just prior to the first scan line of each character row. **DADD9/DW** **Double Width:** Output which denotes a double width character row. **DADD10/UL** **Underline:** Asserted during the blanking interval just prior to the scan line which matches the programmed underline position (lines 0 through 15). **DADD11/BLINK** **Blink Frequency:** Provides an output divided down from the vertical sync rate. **DADD12/ODD** **Odd Field:** Active high signal which is asserted before each scan line of the odd field when interlace is specified. Replaces DADD4/LA0 as the least significant line address for interlaced sync and video applications. **DADD13/LL** **Last Line:** Asserted during the blanking interval just prior to the last scan line of each character row.
V_{CC}	40	44	I	**Power Supply:** +5V power input.
GND	20	22	I	**Ground:** Signal and power ground input.

FUNCTIONAL DESCRIPTION

As shown in the block diagram, the AVDC contains the following major blocks:

- Data bus buffer
- Interface Logic
- Operation Control
- Timing
- Display Control
- Buffer Control

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data buses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the AVDC.

Interface Logic

The interface logic contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are shown in Table 1.

Operation Control

The operation control section decodes configuration and operation commands from the CPU and generates appropriate signals to other internal sections to control the overall device operation. It contains the timing and display registers which configure the display format and operating mode, the interrupt logic, and the status register which provides operational feedback to the CPU.

Timing

The timing section contains the counters and decoding logic necessary to generate the monitor timing outputs and to control the display format. These timing parameters are selected by programming of the initialization registers.

Display Control

The display control section generates linear addressing for up to 16k bytes of display memory. Internal comparators limit the portion of the memory which is displayed to programmed values. Additional functions performed in this section include cursor positioning and address comparisons required for generation of timing signals, double-height tops and bottoms, smooth scrolling, and the split screen interrupts.

Buffer Control

The buffer control section generates three signals which control the transfer of data between the CPU and the display buffer memory. Four system configurations requiring four different 'handshaking' schemes are supported. These are described below.

SYSTEM CONFIGURATIONS

Figure 1 illustrates the block diagram of a typical display terminal using the Signetics SCN2670, SCN2671, SCN2674, and SCB2675 CRT terminal devices. In this system, the CPU examines inputs from the data communications line and the keyboard and places the data to be displayed in the display buffer memory. This buffer is typically a RAM which holds the data for a single or multiple screenload (page) or for a single character row.

The AVDC supports four common system configurations of display buffer memory: the independent, transparent, shared, and row buffer modes. The first three modes utilize a single or multiple page RAM and differ primarily in the means used to transfer display data between the RAM and the CPU. The row buffer mode makes use of a single row buffer (which can be a shift register or a small RAM) that is updated in real time to contain the appropriate display data.

The user programs bits 0 and 1 of IR0 to select the mode best suited for the system environment. The CNTRL1 – 3 outputs perform different functions for each mode and are named accordingly in the description of each mode.

Independent Mode

The CPU to RAM interface configuration for this mode is illustrated in Figure 2. Transfer of data between the CPU and display memory is accomplished via a bidirectional latched port and is controlled by read data buffer ($\overline{\text{RDB}}$), write data buffer ($\overline{\text{WDB}}$), and buffer chip enable ($\overline{\text{BCE}}$). This mode provides a noncontention type of operation that does not require address multiplexers. The CPU does not address the memory directly — the read or write operation is performed at the address contained in the cursor address register or the pointer address register as specified by the CPU. The AVDC enacts the data transfers during blanking intervals in order to prevent visual disturbances of the displayed data.

Table 1. AVDC Addressing

A2	A1	A0	READ ($\overline{\text{RD}}$ = 0)	WRITE ($\overline{\text{WR}}$ = 0)
0	0	0	Interrupt register	Initialization registers[1]
0	0	1	Status register	Command register
0	1	0	Screen start 1 lower register	Screen start 1 lower register
0	1	1	Screen start 1 upper register	Screen start 1 upper register
1	0	0	Cursor address lower register	Cursor address lower register
1	0	1	Cursor address upper register	Cursor address upper register
1	1	0	Screen start 2 lower register	Screen start 2 lower register
1	1	1	Screen start 2 upper register	Screen start 2 upper register

NOTE:

[1] There are 15 initialization registers which are accessed sequentially via a single address. The AVDC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR14) is accessed. The pointer then continues to point to IR14 for additional accesses. Upon a power-on or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command.

The CPU manages the data transfers by supplying commands to the AVDC. The commands used are:

1. Read/write at pointer address.
2. Read/write at cursor address (with optional increment of address).
3. Read/write from cursor address to pointer address.

The operational sequence for a write operation is:

1. CPU checks RDFLG status bit to assure that any delayed commands have been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes address into cursor or pointer registers.
4. CPU issues 'write at cursor with/without increment' or 'write at pointer' command.
5. AVDC generates control signals and outputs specified address to perform requested operation. Data is copied from the interface latch into the memory.
6. AVDC sets RDFLG status to indicate that the write is completed.

Similarly, a read operation proceeds as follows:

1. Steps 1 and 3 as above.
2. CPU issues 'read at cursor with/without increment' or 'read at pointer' command.
3. AVDC generates control signals and outputs specified address to perform requested operation. Data is copied from memory to the interface latch and AVDC sets RDFLG status to indicate that the read is completed.
4. CPU checks RDFLG status to see if operation is completed.
5. CPU reads data from interface latch.

Loading the same data into a block of display memory is accomplished via the 'write from cursor to pointer' command:

1. CPU checks RDFLG status bit to assure that any delayed commands have been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes beginning address of memory block into cursor address register and ending address of block into pointer address register.
4. CPU issues 'write from cursor to pointer' command.
5. AVDC generates control signals and outputs block addresses to copy data from the interface latch into the specified block of memory.
6. AVDC sets RDFLG status to indicate that the block write is completed.

Similar sequences can be implemented on an interrupt driven basis using the READY interrupt output to advise the CPU that a previously asserted delayed command has been completed.

Two timing sequences are possible for the 'read/write at cursor/pointer' commands. If the command is given during the active display window (defined as first scan line of the first character row to the last scan line of the last character row), the operation takes place during the next horizontal blanking interval, as illustrated in Figure 3. If the command is given during the vertical blanking interval, or while the display has been commanded blanked, the operation takes place immediately. In the latter case, the execution time for the command is approximately five character clocks (see Figure 4).

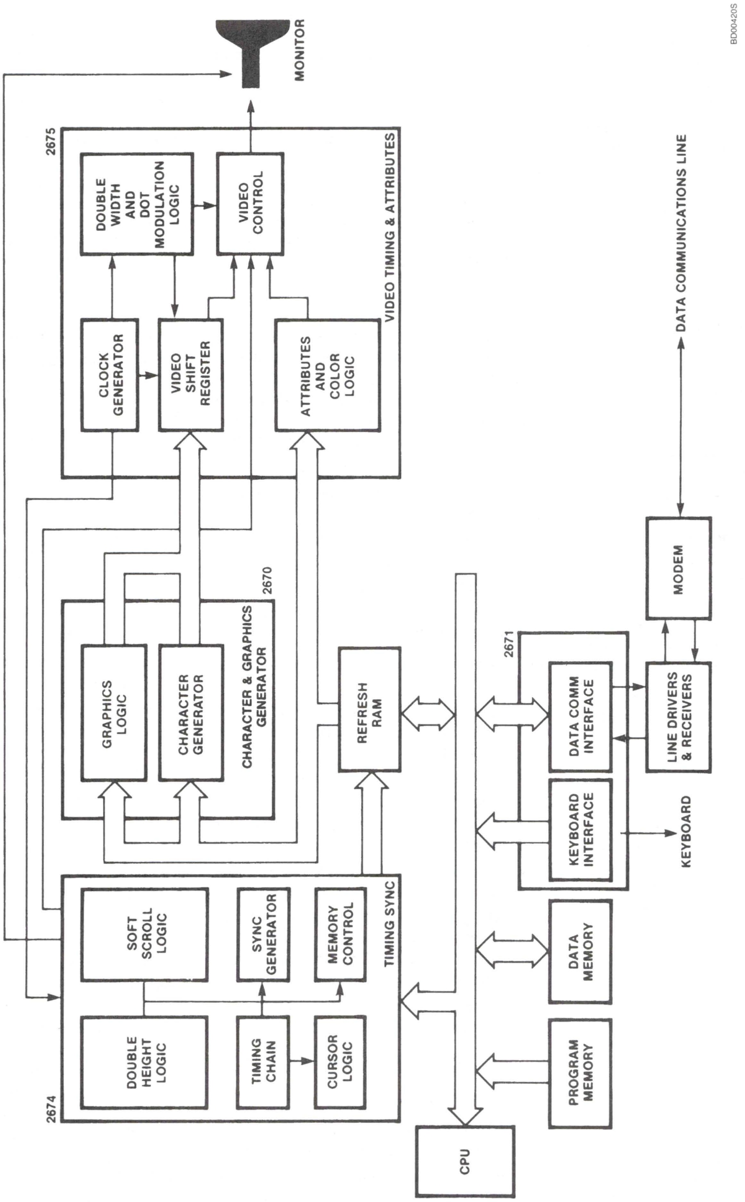

MONITOR
2675
DOUBLE WIDTH AND DOT MODULATION LOGIC
VIDEO CONTROL
CLOCK GENERATOR
VIDEO SHIFT REGISTER
ATTRIBUTES AND COLOR LOGIC
VIDEO TIMING & ATTRIBUTES
DATA COMMUNICATIONS LINE
2670
GRAPHICS LOGIC
CHARACTER GENERATOR
CHARACTER & GRAPHICS GENERATOR
REFRESH RAM
MODEM
2671
DATA COMM INTERFACE
KEYBOARD INTERFACE
LINE DRIVERS & RECEIVERS
KEYBOARD
2674
SOFT SCROLL LOGIC
SYNC GENERATOR
MEMORY CONTROL
DOUBLE HEIGHT LOGIC
TIMING CHAIN
CURSOR LOGIC
TIMING SYNC
DATA MEMORY
PROGRAM MEMORY
CPU
BD00420S

Figure 1. CRT Terminal Block Diagram

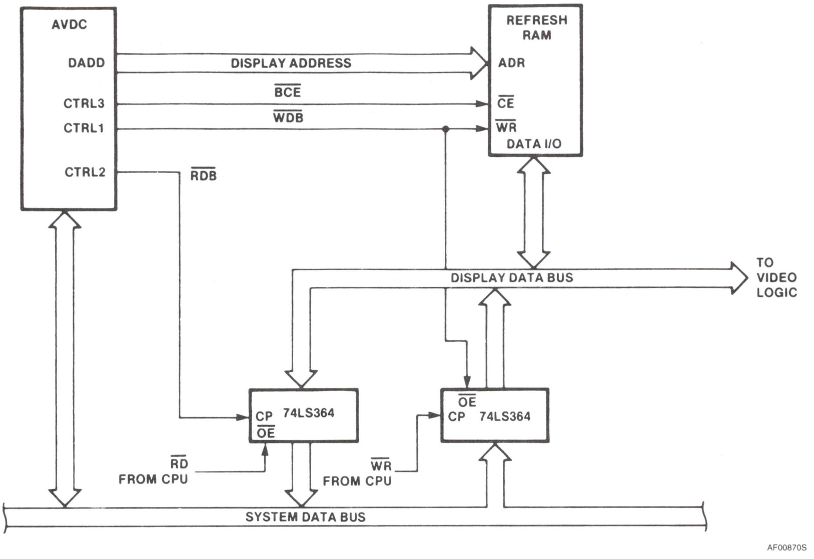
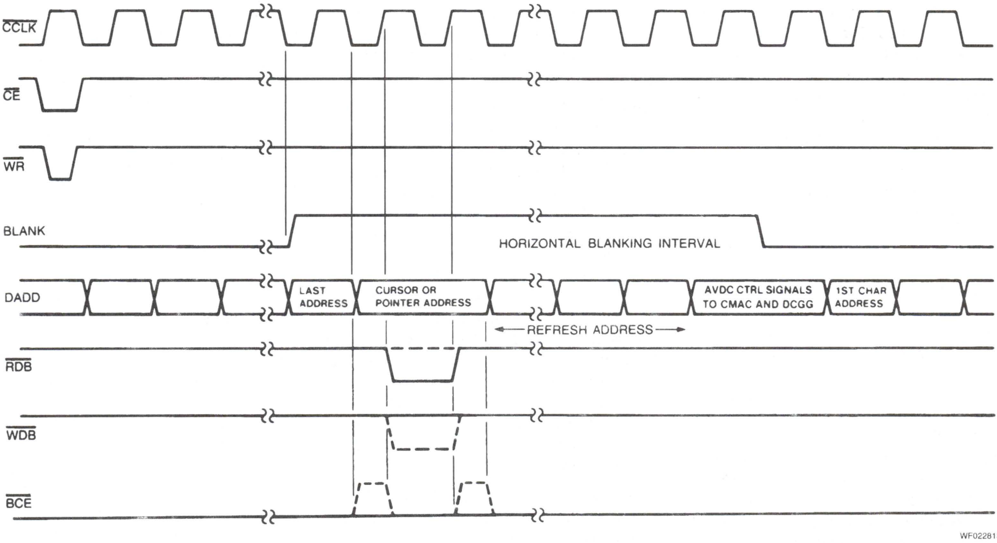

AVDC
DADD
CTRL3
CTRL1
CTRL2
DISPLAY ADDRESS
$\overline{BCE}$
$\overline{WDB}$
$\overline{RDB}$
REFRESH RAM
ADR
$\overline{CE}$
$\overline{WR}$
DATA I/O
DISPLAY DATA BUS
TO VIDEO LOGIC
CP 74LS364
$\overline{OE}$
$\overline{RD}$ FROM CPU
$\overline{WR}$ FROM CPU
SYSTEM DATA BUS
AF00870S

Figure 2. Independent Buffer Mode Configuration


$\overline{CCLK}$
$\overline{CE}$
$\overline{WR}$
BLANK
HORIZONTAL BLANKING INTERVAL
DADD
LAST ADDRESS
CURSOR OR POINTER ADDRESS
REFRESH ADDRESS
AVDC CTRL SIGNALS TO CMAC AND DCGG
1ST CHAR ADDRESS
$\overline{RDB}$
$\overline{WDB}$
$\overline{BCE}$
WF02281S

NOTES:

1. Write waveforms shown in dotted lines.
2. If command execution occurs just prior to the first scan line of a character row and row table addressing mode is enabled, execution of the command is delayed by two character clocks from the timing illustrated

Figure 3. Read/Write at Cursor/Pointer Command Timing (Command Received During Active Display Window)

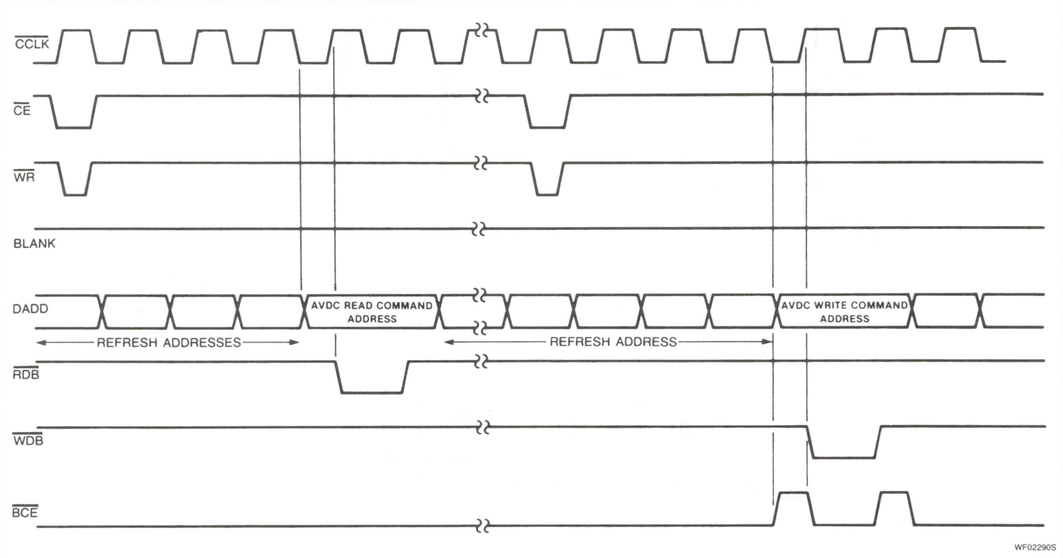


Figure 4. Read/Write at Cursor/Pointer Command Timing (Command Received While Display is Blanked)

Timing for the 'read/write from cursor to pointer' operation is shown in Figure 5. The memory is filled at a rate of one location per two character times. The command will execute only during blanking intervals and may require many horizontal or vertical blanking intervals to complete. Additional delayed commands can be asserted immediately after this command has completed.

Immediate commands can be asserted at any time regardless of the state of the ready status/interrupt.

Shared and Transparent Buffer Modes

In these modes, the display buffer RAM is a part of the CPU memory domain and is addressed directly by the CPU. Both modes use the same hardware configuration with the CPU accessing the display buffer via 3-State drivers (see Figure 6). The processor bus request (PBREQ) control signal informs the AVDC that the CPU is requesting access to the display buffer. In response to this request, the AVDC raises bus acknowledge (BACK) until its bus external (BEXT) output has freed the display address and data buses for CPU access. BACK, which can be used as a 'hold' input to the CPU, is then lowered to indicate that the CPU can access the buffer.

In transparent mode, the AVDC delays the granting of the buffer to the CPU until a vertical or horizontal blanking interval, thereby causing minimum disturbance of the display. In shared mode, the AVDC will blank the display and grant immediate access to the CPU. Timing for these modes is illustrated in Figures 7, 8, and 9.

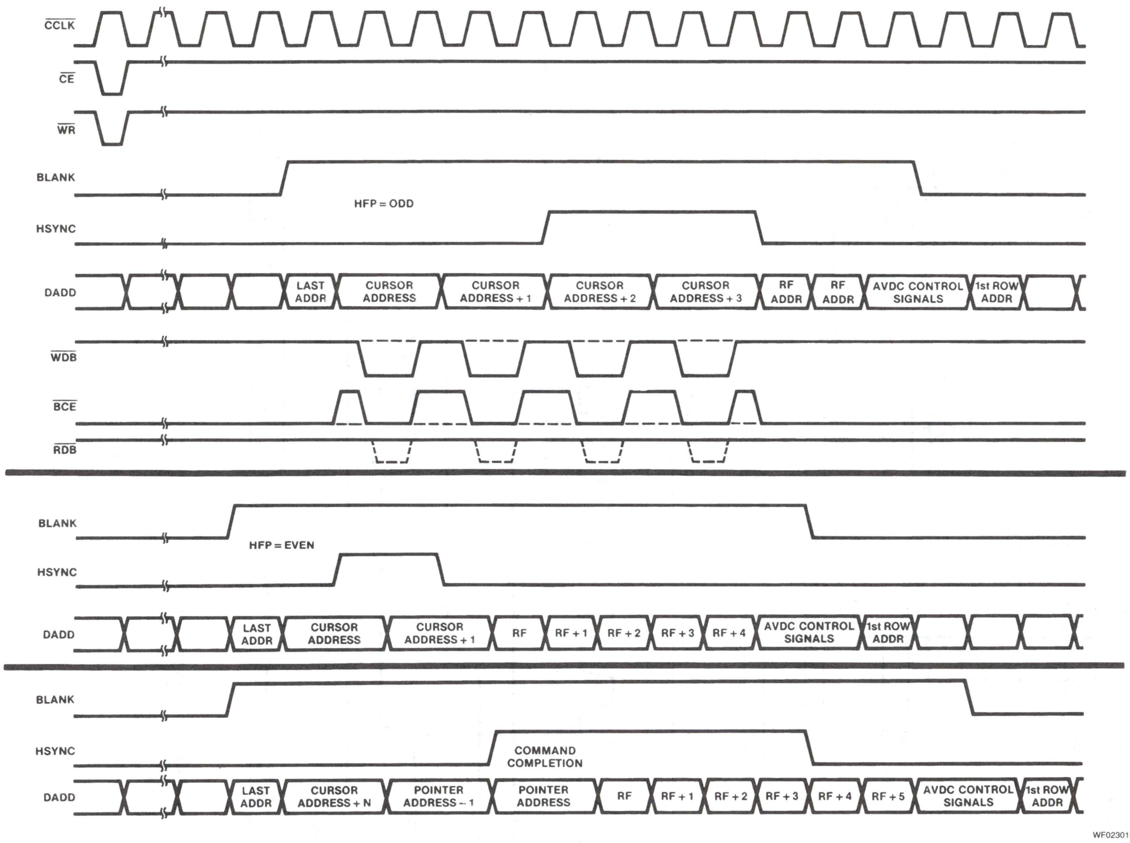
Row Buffer Mode

Figures 10 and 11 show the timing and a typical hardware implementation for the row buffer mode. During the first scan line (line 0) of each character row, the AVDC halts the CPU and DMAs the next row of character data from the system memory to the row buffer memory. The AVDC then releases the CPU and displays the row buffer data for the programmed number of scan lines. The control signal BREQ informs the CPU that character addresses and the MBC signal will start at the next falling edge of BLANK. The CPU must release the address and data buses before this time to prevent bus contention. After the row of character data is transferred to the row buffer RAM, BREQ returns high to grant memory control back to the CPU.

Row Table Addressing Mode

In this mode, each character row in the screen image memory has a unique starting address. This provides greater flexibility with respect to screen operations, such as editing, than the sequential addressing mode. The row table, Figure 12, is a list of starting addresses for each character row and may reside anywhere in the AVDC's addressable memory space. Each entry in the table consists of two bytes: the first byte contains the 8 LSBs of the row starting address and the second byte contains, in its 6 least significant bits, the 6 MSBs of the row starting address. The function of the two MSBs of the second byte is selected by programming IR0[7]. They may be used either as row attribute bits to control double width and double height for that character row, or as an additional two address bits to extend the usable display memory to 64k.

The first address of the row table is designated in screen start register two (SSR2). If row table addressing is enabled via IR2[7], and the display is on, the AVDC fetches the next row's starting address from the table during the blanking interval prior to the first scan line of each character row, while simultaneously incrementing the contents of SSR2 by two so as to point to the next table entry. The fetching of the row starting address from the row table is indicated by the assertion of the CURSOR output during BLANK. The address read from the table by the AVDC is loaded into screen start register 1 (SSR1) for use internally. Since the contents of SSR2 changes as the table entries are fetched, it must be re-initialized to point to the first table entry during each vertical retrace interval.


CCLK
CE
WR
BLANK
HFP = ODD
HSYNC
DADD
LAST ADDR
CURSOR ADDRESS
CURSOR ADDRESS + 1
CURSOR ADDRESS + 2
CURSOR ADDRESS + 3
RF ADDR
RF ADDR
AVDC CONTROL SIGNALS
1st ROW ADDR
WDB
BCE
RDB
BLANK
HFP = EVEN
HSYNC
DADD
LAST ADDR
CURSOR ADDRESS
CURSOR ADDRESS + 1
RF
RF + 1
RF + 2
RF + 3
RF + 4
AVDC CONTROL SIGNALS
1st ROW ADDR
BLANK
HSYNC
COMMAND COMPLETION
DADD
LAST ADDR
CURSOR ADDRESS + N
POINTER ADDRESS – 1
POINTER ADDRESS
RF
RF + 1
RF + 2
RF + 3
RF + 4
RF + 5
AVDC CONTROL SIGNALS
1st ROW ADDR
WF02301S

NOTES:

1. If command execution occurs just prior to the first scan line of a character row and row table addressing mode is enabled, execution of the command is delayed by two character clocks from the timing illustrated.
2. Read waveforms shown in dotted line.

Figure 5. Read/Write From Cursor to Pointer Command Timing

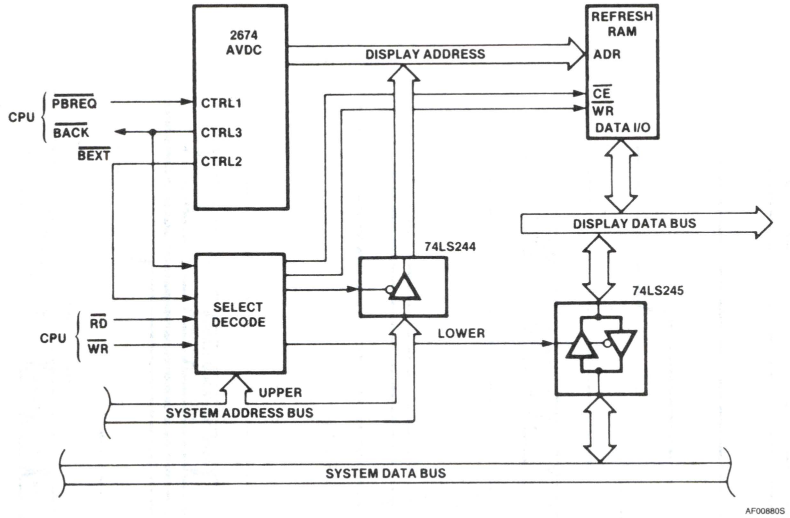
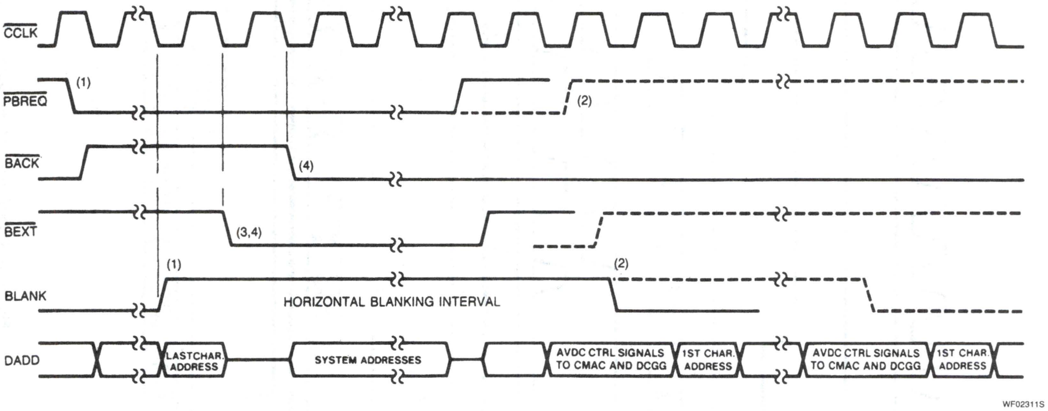
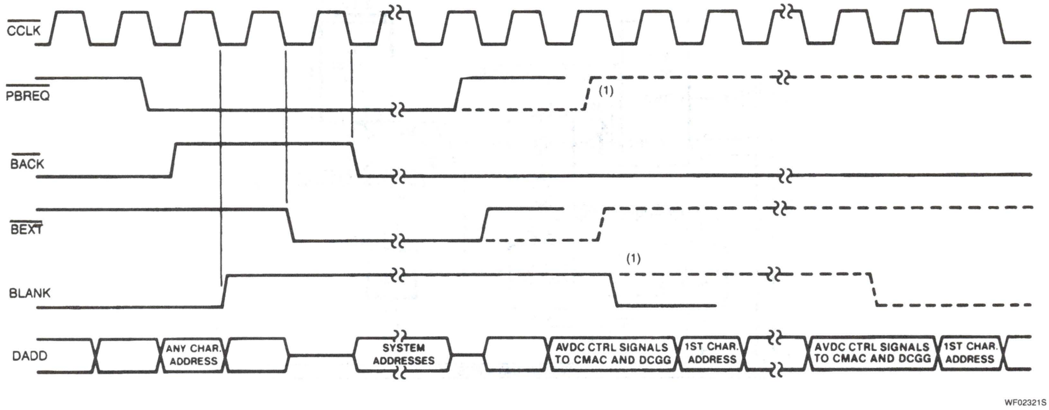

2674
AVDC
CPU
PBREQ
BACK
BEXT
CTRL1
CTRL3
CTRL2
DISPLAY ADDRESS
REFRESH RAM
ADR
CE
WR
DATA I/O
DISPLAY DATA BUS
74LS244
74LS245
SELECT DECODE
RD
WR
LOWER
UPPER
SYSTEM ADDRESS BUS
SYSTEM DATA BUS
AF00880S

Figure 6. AVDC Shared or Transparent Buffer Modes


CCLK
PBREQ
BACK
BEXT
BLANK
DADD
(1)
(2)
(4)
(3,4)
HORIZONTAL BLANKING INTERVAL
LASTCHAR. ADDRESS
SYSTEM ADDRESSES
AVDC CTRL SIGNALS TO CMAC AND DCGG
1ST CHAR. ADDRESS
WF02311S

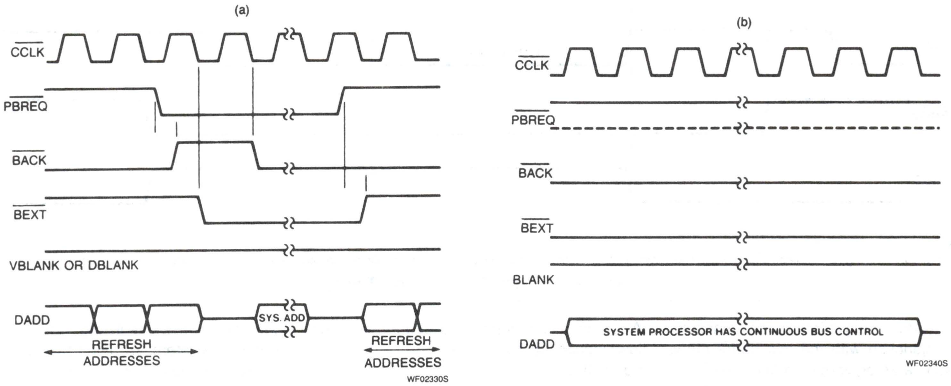
NOTES:
1. PBREQ must be asserted prior to the rising edge of BLANK in order for sequence to begin during that blanking period.
2. If PBREQ is negated after the next to last CCLK of the horizontal blanking interval, the next scan line will also be blanked.
3. Accesses during vertical blank or "display off" are granted only at the beginning of the horizontal front porch.
4. If row table addressing is enabled, CPU access is delayed by two character clocks prior to the first scan line of each character row.

Figure 7. Transparent Buffer Mode Timing


CCLK
PBREQ
BACK
BEXT
BLANK
DADD
(1)
(1)
ANY CHAR. ADDRESS
SYSTEM ADDRESSES
AVDC CTRL SIGNALS TO CMAC AND DCGG
1ST CHAR. ADDRESS
AVDC CTRL SIGNALS TO CMAC AND DCGG
1ST CHAR. ADDRESS
WF02321S

NOTE:
1. If $\overline{\text{PBREQ}}$ is negated after the next to last $\overline{\text{CCLK}}$ of the horizontal blanking interval, the next scan line will also be blanked.

Figure 8. Shared Buffer Mode Timing


(a)
CCLK
PBREQ
BACK
BEXT
VBLANK OR DBLANK
DADD
SYS. ADD
REFRESH ADDRESSES
REFRESH ADDRESSES
WF02330S
(b)
CCLK
PBREQ
BACK
BEXT
BLANK
DADD
SYSTEM PROCESSOR HAS CONTINUOUS BUS CONTROL
WF02340S

a. During Vertical Blank or After 'Display Off' Command in Shared Mode Only. See Figure 7 for Transparent Timing.

b. After 'Display Off and 3-State' Command

Figure 9. Shared and Transparent Mode Timing

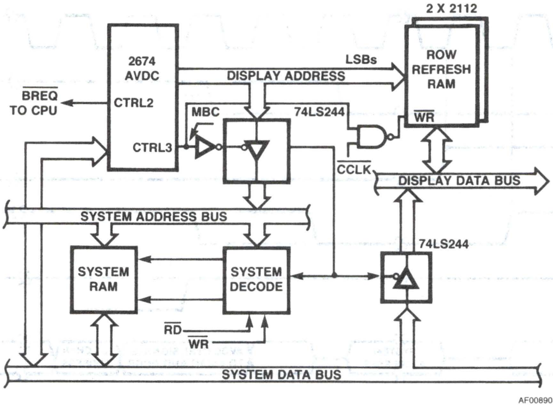
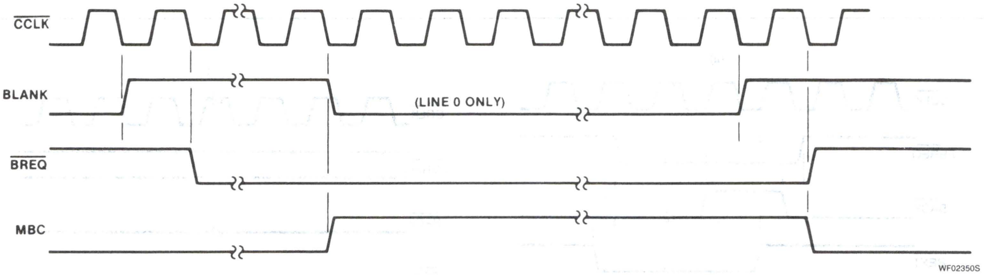


Figure 10. Row Buffer Mode Configuration



NOTE:
If row table addressing is enabled, BREQ will be asserted at the middle of the last scan line of the prior row, and MBC will be asserted at the beginning of BLANK.

Figure 11. Row Buffer Mode Timing

Row table addressing is intended primarily for use in conjunction with the row buffer mode of operation and requires no additional circuitry in that case. It may also be used with the independent and transparent buffer modes, but circuitry must be added to route the data from the display memory to the data bus inputs of the AVDC. Additionally, when not operating in row buffer mode, care must be taken to assure that the CPU does not attempt to access the AVDC while it is reading the row table. One way of preventing this is to latch the last line output which is multiplexed on DADD13 and to test this latch prior to reading or writing the AVDC. The AVDC should only be accessed if the latch is low, indicating that the last line of the row is not active.

Figure 13 illustrates a typical hardware implementation for use in conjunction with independent and transparent modes, and Figure 14 shows the timing for row table operation.

OPERATION

After power is applied, the AVDC will be in an inactive state. Two consecutive 'master reset' commands are necessary to release this circuitry and ready the AVDC for operation. Two register groups exist within the AVDC: the initialization registers and the display control registers. The initialization registers select the system configuration, monitor timing, cursor shape, display memory domain, pointer address, scrolling region, double height and width condition, and screen format. These are loaded first and normally require no modification except for certain special visual effects. The display control registers specify the memory address of the base character (upper left corner of screen), the cursor position, and the split screen addresses associated with the scrolling area or an alternate memory. These may require modification during operation.

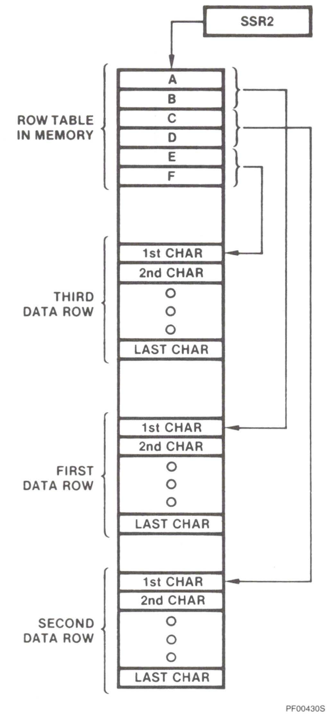


Figure 12. Row Table Address Format

After initial loading of the two register groups, the AVDC is ready to control the monitor screen. Prior to executing the AVDC commands which turn on the display and cursor, the user should load the display memory with the first data to be displayed. During operation, the AVDC will sequentially address the display memory within the limits programmed into its registers. The memory outputs character codes to the system character and graphics generation logic, where they are converted to the serial video stream necessary to display the data on the CRT. The user effects changes to the display by modifying the contents of the display memory, the AVDC display control and command registers, and the initialization registers, if required. Interrupts and status conditions generated by the AVDC supply the 'handshaking' information necessary for the CPU to effect real-time display changes in the proper time frame if required.

Initialization Registers

There are 15 initialization registers (IR0 – IR14) which are accessed sequentially via a single address. The AVDC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR14) is accessed. The pointer then continues to point to IR14 for further accesses. Upon a power-on or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command. These registers are write only and are used to specify parameters such as the system configuration, display format, cursor shape, and monitor timing. Register formats are shown in Table 2.

IR0[7] — Double Height/Width Enable

When this bit is set, the value in IR14[7:6] is used to control the double height and width conditions of each character row. Assertion of this bit also allows IR14[7:6] to be programmed in two ways:

1. By the CPU writing to IR14 directly.
2. When the contents of screen start register 1 (SSR1) upper are changed, either by the CPU writing to this register or by the automatic loading of SSR1 when operating in row table mode, the two MSBs of SSR1 upper are copied into IR14[7:6]. Thus, the MSBs of each row table entry can be used to control double height and double width attributes on a row by row basis.

IR14[5:4] are not active when this bit is set. When this bit is reset, the double height and width attributes operate as described in IR[14].

IR0[6:3] — Scan Lines Per Character Row

Both interlaced and non-interlaced scanning are supported by the AVDC. For interlaced mode, two different formats can be implemented, depending on the interconnection between the AVDC and the character generator (see IR1[7]). This field defines the number of scan lines used to compose a character row for each technique. As scanning occurs, the scan line count is output on the LA0 – LA3 and ODD pins.

IR0[2] — VSYNC/CSYNC Enable

This bit selects either vertical sync pulses or composite sync pulses on the VSYNC/CSYNC output (pin 18). The composite sync waveform conforms to EIA RS-170 standards, with the vertical interval composed of six equalizing pulses, six vertical sync pulses, and six more equalizing pulses.

IR0[1:0] — Buffer Mode Select

Four buffer memory modes may be selectively enabled to accommodate the desired system configuration. (See System Configurations).

IR1[7] — Interlace Enable

Specifies interlaced or non-interlaced timing operation. Two modes of interlaced operation are available, depending on whether LA0 – LA3 or ODD, LA0 – LA2 are used as the line address for the character generator. The resulting displays are shown in Figure 15.

For 'interlaced sync' operation, the same information is displayed in both odd and even fields, resulting in enhanced readability. The AVDC outputs successive line numbers in ascending order on the LA0 – LA3 lines, one per scan line for each field.

The 'interlaced sync and video' format doubles the character density on the screen. The AVDC outputs successive line numbers in ascending order on the ODD and LA0 – LA2 lines, one per scan line for each field. The number of scan lines per character row is always even. Assume that the first character row is row 0 (even). When scanning through the odd field, the scan line numbers being displayed are odd for both the even and odd character rows. When scanning through the even field, the scan line numbers being displayed are even for both even and odd character rows (see Figure 15).

IR1[6:0] — Equalizing Constant

This field indirectly defines the horizontal front porch and is used internally to generate the equalizing pulses for the RS-170 compatible CSYNC. The value for this field is the total number of character clocks ($\overline{\text{CCLKs}}$) during a horizontal line period divided by two, minus two times the number of character clocks in the horizontal sync pulse:

$$EC = \frac{H_{ACT} + H_{FP} + H_{SYNC} + H_{BP}}{2} - 2(H_{SYNC})$$

The definition of the individual parameters is illustrated in Figure 16. The minimum value of H_{FP} is three character clocks, four $\overline{\text{CCLK}}$s for row table addressing. Note that when using the 2675 CMAC, it will delay the blank pulse three $\overline{\text{CCLK}}$s relative to the HSYNC pulse. Because of this delay, the actual HFP and HBP values will be different from the values programmed into the AVDC. The actual HFP will be decreased by 3 character clocks. The actual HBP will be increased by 3 character clocks.

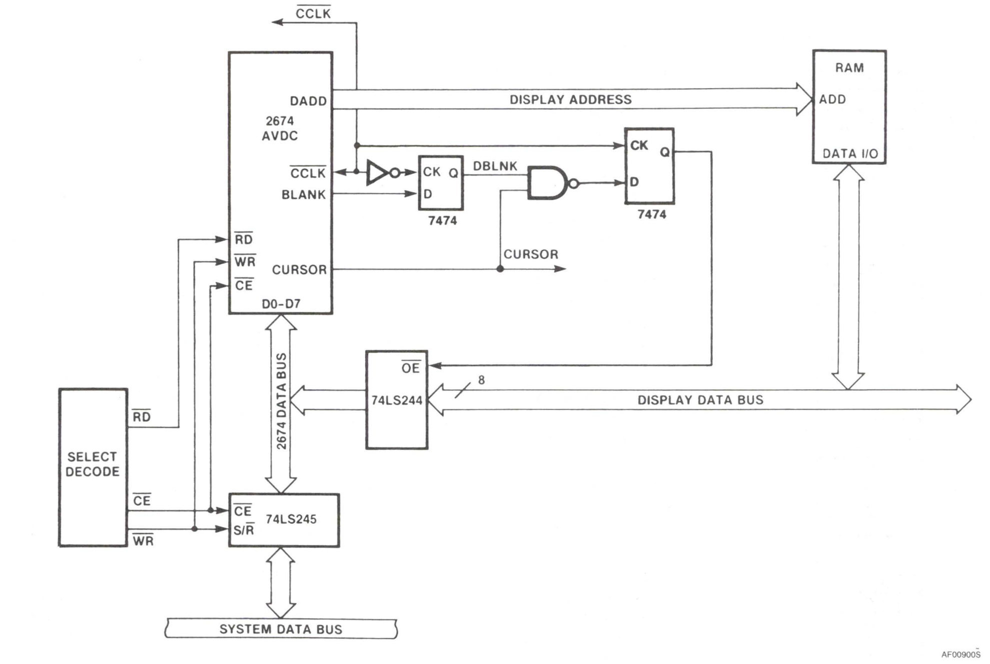


Figure 13. Row Table Mode Configuration (Non-Row Buffer Modes)

IR2[7] — Row Table Mode Enable
Assertion/negation of this bit causes the AVDC to begin/terminate operating in row table mode starting at the next character row. See Row Table Addressing Mode section. By using the split interrupt capability of the AVDC, this mode can be enabled and disabled on a particular character row. This allows a combination of row table and sequential addressing to be utilized to provide maximum flexibility in generating the display.

IR2[6:3] — Horizontal Sync Pulse Width
This field specifies the width of the HSYNC pulse in $\overline{\text{CCLK}}$ periods.

IR2[2:0] — Horizontal Back Porch
This field defines the number of $\overline{\text{CCLK}}$s between the trailing edge of HSYNC and the trailing edge of BLANK.

IR3[7:5] — Vertical Front Porch
Specifies the number of scan line periods between the rising edges of BLANK and VSYNC during the vertical retrace interval. The vertical front porch will be extended in increments of scan lines if the ACLL input is low at the end of the programmed value.

IR3[4:0] — Vertical Back Porch
This field determines the number of scan line periods between the falling edges of the VSYNC and BLANK outputs.

IR4[7] — Character Blink Rate
Specifies the frequency for the character blink attribute timing. The blink rate can be specified as 1/64 or 1/128 of the vertical field rate. The timing signal has a duty cycle of 50% and is multiplexed onto the DADD11/BLINK output at the falling edge of each BLANK.

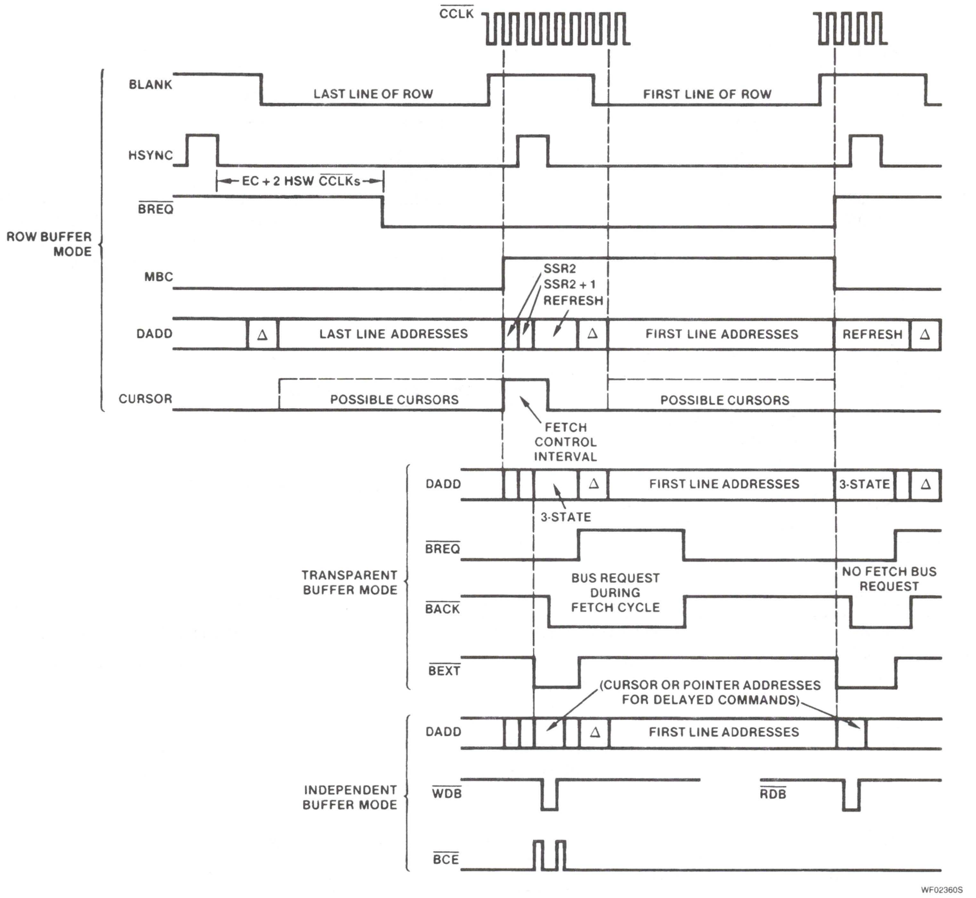
IR4[6:0] — Character Rows Per Screen
This field defines the number of character rows to be displayed. This value multiplied by the scan lines per character row, plus the vertical front porch, the vertical back porch values, and the vertical sync pulse width is the vertical scan period in scan lines.

IR5[7:0] — Active Characters Per Row
This field determines the number of characters to be displayed on each row of the CRT screen. The sum of this value, the horizontal front porch, the horizontal sync width, and the horizontal back porch is the horizontal scan period in $\overline{\text{CCLK}}$s.

IR6[7:4], IR6[3:0] — First and Last Scan Line of Cursor
These two fields specify the height and position of the cursor on the character block. The 'first' line is the topmost line when scanning from the top to the bottom of the screen. The value of the 'first line of cursor' must be less than the 'last line of cursor' value.

IR7[7:6] — Vertical Sync Pulse Width
This field specifies the width of the VSYNC pulse in scan line periods.

IR7[5] — Cursor Blink Enable
This bit controls whether or not the cursor output pin will be blinked at the selected rate (IR7[4]). The blink duty cycle for the cursor is 50%.


CCLK
BLANK
LAST LINE OF ROW
FIRST LINE OF ROW
HSYNC
EC + 2 HSW CCLKs
BREQ
ROW BUFFER MODE
MBC
SSR2
SSR2 + 1
REFRESH
DADD
Δ
LAST LINE ADDRESSES
FIRST LINE ADDRESSES
REFRESH
CURSOR
POSSIBLE CURSORS
FETCH CONTROL INTERVAL
3-STATE
BUS REQUEST DURING FETCH CYCLE
NO FETCH BUS REQUEST
TRANSPARENT BUFFER MODE
BACK
BEXT
(CURSOR OR POINTER ADDRESSES FOR DELAYED COMMANDS)
INDEPENDENT BUFFER MODE
WDB
RDB
BCE
WF02360S

NOTES:
Δ = Multiplexed control signals (LG,DW, etc.)
EC = Equalizing constant
HSW = Horizontal sync width

Figure 14. Row Table Mode Timing

Table 2. Initialization Register Bit Formats

	BIT 7	BIT 6 BIT 5	BIT 4 BIT 3	BIT 2	BIT 1 BIT 0
	DOUBLE HT/WD	SCAN LINES PER CHARACTER ROW		SYNC SELECT	BUFFER MODE SELECT
		NON-INTERLACED	INTERLACED		
IR0	0 = OFF 1 = ON	0000 = 1 LINE 0001 = 2 LINES 0010 = 3 LINES • • 1110 = 15 LINES 1111 = 16 LINES	0000 = 2 LINES 0001 = 4 LINES 0010 = 6 LINES • • 1110 = 30 LINES 1111 = UNDEFINED	0 = VSYNC 1 = CSYNC	00 = INDEPENDENT 01 = TRANSPARENT 10 = SHARED 11 = ROW

	BIT 7	BIT 6 BIT 5 BIT 4 BIT 3 BIT 2 BIT 1 BIT 0
	INTERLACE ENABLE	EQUALIZING CONSTANT
IR1	0 = NON–INT 1 = INTER	00000000 = 1 CCLK 00000001 = 2 CCLK • • 11111110 = 127 CCLK 11111111 = 128 CCLK CALCULATED FROM: $EC = 0.5(H_{ACT} + H_{FP} + H_{SYNC} + H_{BP}) - 2(H_{SYNC})$

	BIT 7	BIT 6 BIT 5 BIT 4 BIT 3	BIT 2 BIT 1 BIT 0
	ROW TABLE	HORIZONTAL SYNC WIDTH	HORIZONTAL BACK PORCH
IR2	0 = OFF 1 = ON	0000 = 2 CCLK 0001 = 4 CCLK • • 1110 = 30 CCLK 1111 = 32 CCLK	000 = NOT ALLOWED 001 = 3 CCLK • • 110 = 23 CCLK 111 = 27 CCLK

	BIT 7 BIT 6 BIT 5	BIT 4 BIT 3 BIT 2 BIT 1 BIT 0
	VERTICAL FRONT PORCH	VERTICAL BACK PORCH
IR3	000 = 4 SCAN LINES 001 = 8 SCAN LINES • • 110 = 28 SCAN LINES 111 = 32 SCAN LINES	00000 = 4 SCAN LINES 00001 = 6 SCAN LINES • • 11110 = 64 SCAN LINES 11111 = 66 SCAN LINES

	BIT 7	BIT 6 BIT 5 BIT 4 BIT 3 BIT 2 BIT 1 BIT 0
	CHARACTER BLINK RATE	ACTIVE CHARACTER ROWS PER SCREEN
IR4	0 = 1/64 VSYNC 1 = 1/128 VSYNC	00000000 = 1 ROW 00000001 = 2 ROWS • • 11111110 = 127 ROWS 11111111 = 128 ROWS

Table 2. Initialization Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR5	ACTIVE CHARACTERS PER ROW							
	00000010 = 3 CHARACTERS 00000011 = 4 CHARACTERS • • 11111110 = 255 CHARACTERS 11111111 = 256 CHARACTERS							

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR6	FIRST LINE OF CURSOR				LAST LINE OF CURSOR			
	0000 = SCAN LINE 0 0001 = SCAN LINE 1 • • 1110 = SCAN LINE 14 1111 = SCAN LINE 15				0000 = SCAN LINE 0 0001 = SCAN LINE 1 • • 1110 = SCAN LINE 14 1111 = SCAN LINE 15			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR7	VSYNC WIDTH		CURSOR BLINK	CURSOR RATE	UNDERLINE POSITION			
	00 = 3 SCAN LN 01 = 1 SCAN LN 10 = 5 SCAN LN 11 = 7 SCAN LN		0 = OFF 1 = ON	0 = 1/32 1 = 1/64	0000 = SCAN LINE 0 0001 = SCAN LINE 1 • • 1110 = SCAN LINE 14 1111 = SCAN LINE 15			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR8	DISPLAY BUFFER FIRST ADDRESS LSB'S							
	H'000 = 0 H'001 = 1 • • H'FFE = 4,094 H'FFF = 4,095				NOTE: MSB'S ARE IN IR9[3:0]			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR9	DISPLAY BUFFER LAST ADDRESS				DISPLAY BUFFER FIRST ADDRESS MSB'S			
	0000 = 1,023 0001 = 2,047 • • 1110 = 15,359 1111 = 16,383				SEE IR8			

Table 2. Initialization Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR10	DISPLAY POINTER ADDRESS LOWER							
	SEE IR11							

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR11	LZ DOWN	LZ UP	DISPLAY POINTER ADDRESS UPPER					
	0 = OFF 1 = ON	0 = OFF 1 = ON	H'0000' = 0 H'0001' = 1 • • H'3FFF' = 16,383					

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR12	SCROLL START	SPLIT REGISTER 1						
	0 = OFF 1 = ON	00000000 = ROW 1 00000001 = ROW 2 • • 11111111 = ROW 128						

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR13	SCROLL END	SPLIT REGISTER 2						
	0 = OFF 1 = ON	00000000 = ROW 1 00000001 = ROW 2 • • 11111111 = ROW 128						

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IR14	DOUBLE 1		DOUBLE 2		LINES TO SCROLL			
	00 = NORMAL 01 = DOUBLE WIDTH 10 = DB WD & TOPS 11 = DB WD & BOTS		00 = NORMAL 01 = DOUBLE WIDTH 10 = DB WD & TOPS 11 = DB WD & BOTS		0000 = 1 SCAN LINE 0 0001 = 2 SCAN LINE 1 • • • • 1110 = 15 SCAN LINE 14 1111 = 16 SCAN LINE 15			

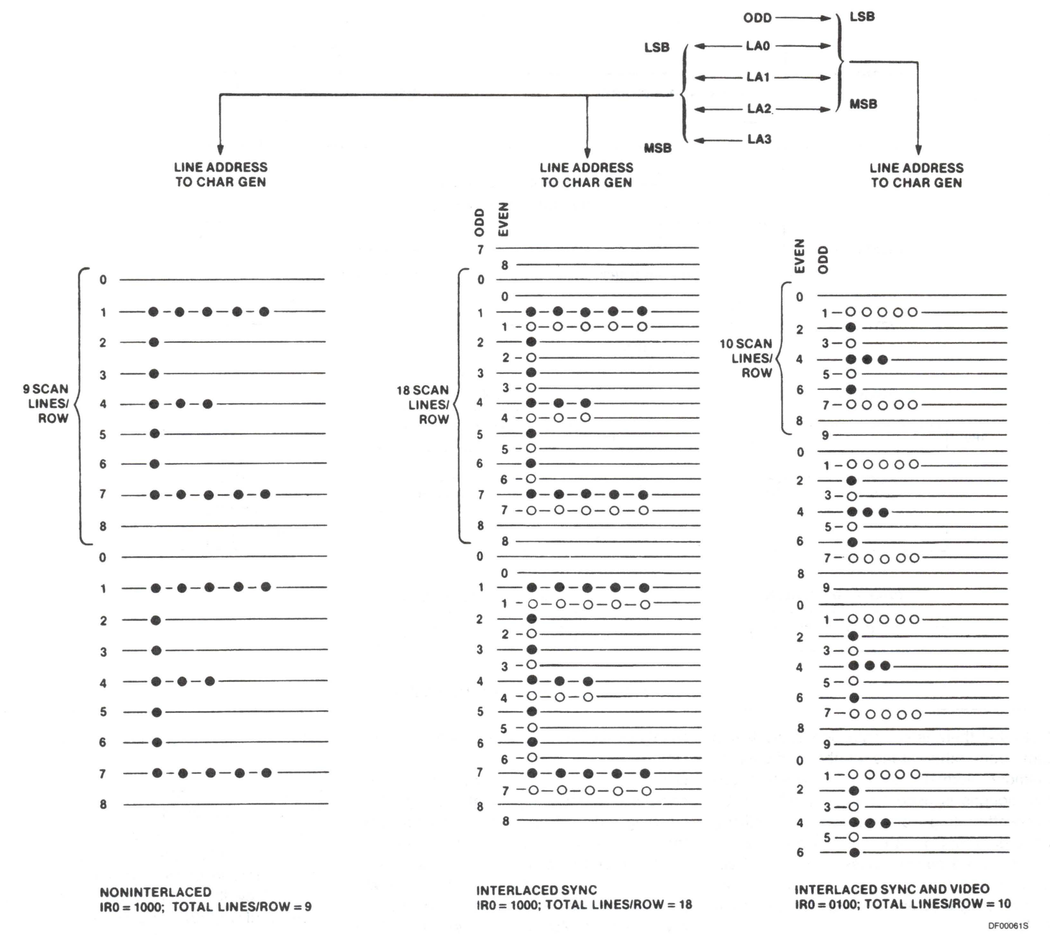

ODD
LSB
LSB
LA0
LA1
LA2
LA3
MSB
MSB
LINE ADDRESS TO CHAR GEN
LINE ADDRESS TO CHAR GEN
LINE ADDRESS TO CHAR GEN
ODD
EVEN
EVEN
ODD
9 SCAN LINES/ ROW
18 SCAN LINES/ ROW
10 SCAN LINES/ ROW
NONINTERLACED
IR0 = 1000; TOTAL LINES/ROW = 9
INTERLACED SYNC
IR0 = 1000; TOTAL LINES/ROW = 18
INTERLACED SYNC AND VIDEO
IR0 = 0100; TOTAL LINES/ROW = 10
DF00061S

Figure 15. Interlaced Display Modes

IR7[4] — Cursor Blink Rate
The cursor blink rate can be specified at 1/32 or 1/64 of the vertical scan frequency. Blink is effective only if blink is enabled by IR7[5].

IR7[3:0] — Underline Position
This field defines which scan line of the character row will be used for the underline attribute by the 2675 CMAC. The timing signal is multiplexed onto the DADD10/UL output during the falling edge of BLANK.

IR9[3:0], IR8[7:0] — Display Buffer First Address

IR9[7:4] — Display Buffer Last Address
These two fields define the area within the buffer memory where the display data will reside. When the data at the 'display buffer last address' is displayed, the AVDC will wraparound and obtain the data to be displayed at the next screen position from the 'display buffer first address'. If 'last address' is the end of a character row and a new screen start address has been loaded into the screen start register, or if 'last address' is the last character position of the screen, the next data is obtained from the address contained in the screen start register.

Advanced Video Display Controller (AVDC) SCN2674

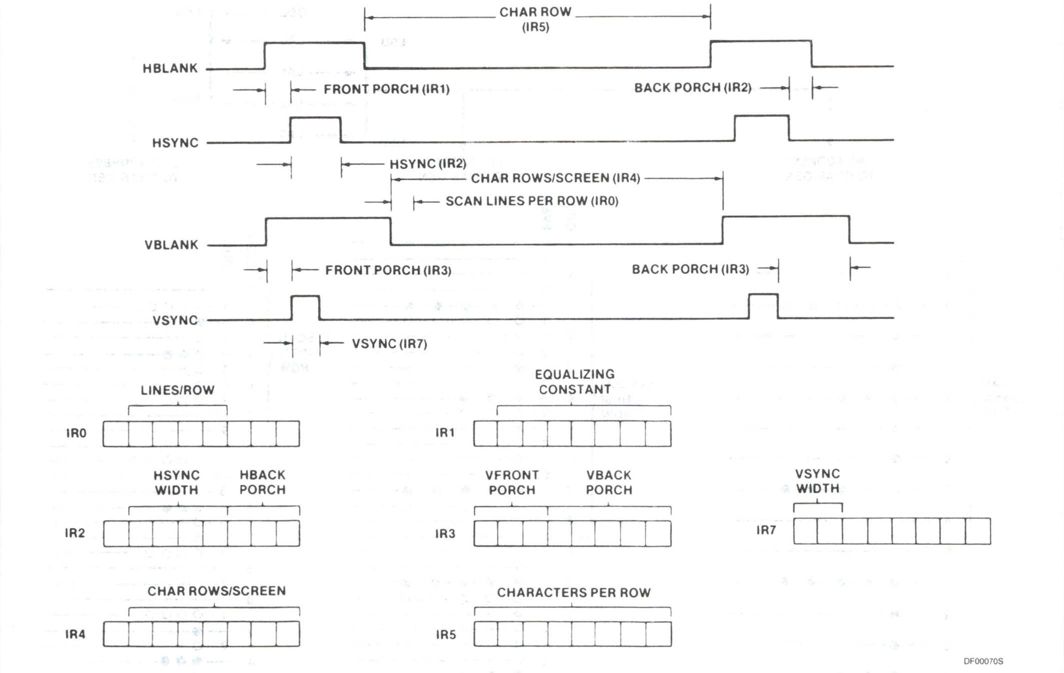


Figure 16. Horizontal and Vertical Timing

Note that there is no restriction in displaying data from other areas of the addressable memory. Normally, the area between these two bounds is used for data which can be overwritten (e.g., as a result of scrolling), while data that is not to be overwritten would be contained outside these bounds and accessed by means of the automatic split screen or split screen interrupt feature of the AVDC.

IR10[7:0] — Display Pointer Address, Lower

IR11[5:0] — Display Pointer Address, Upper

These two fields define a buffer memory address for AVDC controlled accesses in response to 'read/write at pointer' commands. They also define the last buffer memory address to be written for the 'write from cursor to pointer' command.

In the independent mode, the RDFLG bit of the status register should be checked for the ready state (bit 5 equal to a logic one) before writing to the display pointer address registers. Checking the status register will prevent the pointer address from being changed while a delayed command (e.g. write from cursor to pointer) is still being executed.

IR11[7] — Scan Line Zero During Scroll Down

During a scroll down operation, the new character row will appear at the top of the scrolling region. If this bit is set (logic one), the scan line count pins (LA0 through LA3) will be forced to zero for every scan line of the partial row. If the character generator provides blanks for scan line zero, the new row being scrolled into the screen will be blanked. This feature can be used to blank the new row to give the CPU time to load the new data in the display buffer. When this bit is set to a logic zero, the new data will be displayed.

IR11[6] — Scan Line Zero During Scroll Up

During a scroll up operation, the new character row will appear at the bottom of the scrolling region. If this bit is set (logic one), the scan line count pins (LA0 through LA3) will be forced to zero for every scan line of the partial row. If the character generator provides blanks for scan line zero, the new row being scrolled into the screen will be blanked. This feature can be used to blank the new row to give the CPU time to load the new data in the display buffer. When this bit is set to a logic zero, the new data will be displayed.

IR12[7] — Scroll Start

This bit is asserted when soft scroll is to take place. The scrolling area begins at the row specified in split register 1 (IR12[6:0]). If set, the first row to scroll scan line count will be reduced by the value in the lines to scroll register (IR14[3:0]). The scan line count of this row will start at the programmed offset value. When this bit is asserted, scroll end IR13[7] must be set before split 2.

IR12[6:0] — Split Register 1

Split register 1 can be used to provide special screen effects such as soft (scan line by scan line) scrolling, double height/width rows, or to change the normal addressing sequence of the display memory. The contents of this field are compared, in real time, to the current row number. Upon a match, the AVDC sets the split screen 1 status bit, and issues an interrupt request if so programmed. The status change/interrupt request is made at the beginning of scan line zero of the split screen character row. If enabled by the SPL1 bit of screen start register 2, an automatic split screen to the address specified in screen start register 2 will be made for the designated character row. During a scroll operation, this field defines the first character row of the scrolling area.

IR13[7] — Scroll End
This field specifies that the row programmed in split register 2 (IR13[6:0]) is to be the last scrolling row of the scrolling area. Note that this bit must be asserted for a valid row only when the scroll start bit IR12[7] is also asserted.

IR13[6:0] — Split Register 2
This field is similar to the split register 1 field except for the following:

1. Split screen 2 status bit is set.
2. During a scroll operation, this field defines the last character row of the scrolling area. This row will be followed by a partial row. The LTSR (IR14) value replaces the normal scan lines/row value for the partial row, thus keeping the total scan lines/screen the same.
3. If enabled by the SPL2 bit of screen start register 2, an automatic split to the address contained in screen start register 2 will occur in one of two ways: a) If not scrolling an automatic split will occur for the next character row. b) If scrolling, the automatic split will occur after the partial row being scrolled onto or off the screen.
4. The specified double width and height conditions (IR14) are also asserted in two possible ways: a) Automatic split will assert the programmed condition for the current row. b) During soft scroll operation the programmed conditions are asserted for the partial row scrolling onto or off the screen.

IR14[7:6] — Double 1
This field specifies the conditions (double width/height or normal) of the row designated in split register 1 (IR12[6:0]). When double height tops or bottoms have been specified, the AVDC will automatically toggle between tops and bottoms until another split 1 or 2 occurs which changes the double height/width condition. If a double height top row is specified, the scan line count will start at zero and increment the scan line count every other scan line. If a double height bottom row is specified, the AVDC will start at one-half the normal scan line total. If double width is specified, the AVDC will assert the DADD9/DW output at the falling edge of blank. This condition will also remain active until the next split 1 or 2. When IR0[7] = 1, the values written into bits 7 and 6 of screen start 1 upper will also be written into IR14[7:6] and the automatic toggling between tops and bottoms is disabled.

The AVDC still addresses the RAMs on a single width character basis. The clock rate of the AVDC does not change. The display RAMs must have data as if two single-wide characters are to be displayed. The first data bits addressed by the AVDC will specify the double wide character. The next data bits addressed are not displayed. The 2675 CMAC will ignore the second clock cycle data when the ADOUBLE pin is high.

IR14[5:4] — Double 2
This field specifies the conditions (double width/height or normal) of the row designated in split register 2 (IR13[6:0]). Not used when IR0[7] = 1.

IR14[3:0] — Lines To Scroll
This field defines the scan line increment to be used during a soft scroll operation. These 4 bits control the scroll rate. When smooth scrolling up by scan line increments of one, the initial value is 0000 (scan line 0) and is increased every vertical frame according to the number of lines per character row. When smooth scrolling down by scan line increments of one, the initial value is 1110 hex (scan line 14, assuming 15 scan lines per character row) and is decreased by one every vertical frame. This value will only be used when scroll start (IR12[7]) and scroll end (IR13[7]) are enabled.

Timing Considerations
Normally, the contents of the initialization registers are not changed during normal operation. However, this may be necessary to implement special display features such as multiple cursors and horizontal scrolling. Table 3 describes timing details for these registers which should be considered when implementing these features.

Display Control Registers

There are seven registers in this group, each with an individual address. Their formats are illustrated in Table 4. The command register is used to invoke one of 19 possible AVDC commands as described in the COMMANDS section of this data sheet. The remaining registers in the group store address values which specify the cursor location, the location of the first character to be displayed on the screen, and any split screen address locations. The user initializes these registers after powering on the system and changes their values to control the data which is displayed.

Screen Start Registers 1 and 2
The screen start 1 registers contain the address of the first character of the first row (upper left corner of the active display). At the beginning of the first scan line of the first row, this address is transferred to the row start register (RSR) and into the memory address counter (MAC). The counter is then advanced sequentially at the character clock rate for the number of times programmed into the active characters per row register (IR5), thus reaching the address of the last character of the row plus one. At the beginning of each subsequent scan line of the first row, the MAC is reloaded from the RSR and the above sequence is repeated. At the end of the last scan line of the first row, the contents of the MAC are loaded into the RSR to serve as the starting memory address for the second character row. This process is repeated for the programmed number of rows per screen. Thus, the data in the display memory is displayed sequentially starting from the address contained in the screen start register. After the ensuing vertical retrace interval, the contents of the screen start registers are reloaded into the RSR and MAC, and the process is repeated.

During vertical blanking, the address counter operation is modified by stopping the automatic load of the contents of the RSR into the counter, thereby allowing the address outputs to free-run. This allows dynamic memory refresh to occur during the vertical retrace interval. The refresh addressing starts at the last address displayed on the screen and increments by one for each character clock during the retrace interval. If the display buffer last address is encountered, wraparound will occur. Refreshing will continue from the display buffer first address. In the independent mode, the refresh addressing will occur if no delayed commands are being executed. In the transparent and shared modes, refresh will occur during the blanking interval unless the CPU has control of the display address bus. In the row buffer mode, refresh will occur during all blanking intervals except for the first character clock time in the BLANK after the first scan line (scan line 0) of a character row.

The sequential addressing operation described above will be modified upon the occurrence of the following:

1. After reaching the 'display buffer last address.'
2. Rewriting the contents of the screen start 1 registers.
3. Setting the split register 1 or split register 2 bits of screen start register 2 upper.
4. Enabling the row table addressing mode.

First, if during the incrementing of the memory address counter the 'display buffer last address' (IR9[7:4]) is reached, the MAC will be loaded from the 'display buffer first address' register (IR9[3:0] and IR8[7:0]) at the next character clock. Sequential operation will then resume starting from this address. This wraparound operation allows portions of the display buffer to be used for purposes other than storage of displayable data and is completely automatic without any CPU intervention (see Figure 17a).

Second, if the contents of screen start register 1 (upper, lower, or both) are changed during any character row (e.g., row 'n'), the starting address of the next character row

(row 'n + 1') will be the new value of the screen start register and addressing will continue sequentially from there. This allows features such as split screen operation, partial scroll, or status line display to be implemented. The split screen interrupt feature of the AVDC is useful in controlling the CPU initiated operations. Note that in order to obtain the correct screen display, screen start register 1 must be reloaded with the original (origin of display) value prior to the end of the vertical retrace. See Figure 17b.

The screen start two registers contain a 14-bit display address. The SSR2 address is implemented on the occurrence of item 3 above. If bit 6 of SSR2 upper is set, the SSR2 contents will be automatically loaded into the RSR at the beginning of the first scan line of the row designated by split register 1 (IR12[6:0]). If bit 7 of SSR2 upper is set, the SSR2 contents will be automatically loaded into the RSR at the beginning of the first scan line of the row specified by split register 2 (IR13[6:0]). SPL1 and SPL2 are write only bits and will read as zero when reading screen start register 2. If these bits are not used, they should be set to zeros after power-up.

In order to avoid screen start register 1 and 2 (SSR1, SSR2) writing sequence conflict, after SSR1 and/or SSR2 are loaded with new values, SSR2 Value needs to be checked. If SSR2 value is incorrect, multiple SSR2 rewrites may be necessary.

Lastly, when row table addressing mode is enabled, the first address of the row table is designated in SSR2. The AVDC fetches the next row's starting address from the table during the blanking interval prior to the first scan line of each character row and loads it into SSR1 for use as the starting address of the next row. Since the contents of SSR2 change as the table entries are fetched, it must be re-initialized to point to the first table entry during each vertical retrace interval.

The values in the two MSBs of SSR1 upper are multiplexed onto the DADD1/DADD14 and DADD2/DADD15 outputs during the falling edge of BLANK. If IR0[7] = 0, these two bits act as memory page select bits which may be used to extend the display memory addressing range of the AVDC up to 64k. In that case, these two bits act as a two-bit counter which is incremented each time that 'wraparound' occurs (see above). Note that the counter is incremented at the falling edge of BLANK and that for proper display operation the wraparound address should be programmed to occur at the last character position of a row. Also, the first address accessed in the new page will be the address contained in the display buffer first address register (IR9[3:0] and IR8[7:0]). DADD14 and DADD15 should only be used in the bit-mapped graphics mode.

Table 3. Timing Considerations

PARAMETER	TIMING CONSIDERATIONS
First line of cursor Last line of cursor Underline line	These parameters must be established at a minimum of two character times prior to their occurrence
Double height character rows Double width character rows Rows to scroll	Set/reset prior to the row specified in split 1 or 2 registers
Cursor blink Cursor blink rate Character blink rate	New values become effective within one field after values are changed
Split register 1 Split register 2	Change anytime prior to line zero of desired row
Character rows per screen	Change only during vertical blanking period
Vertical front porch	Change prior to first line of VFP
Vertical back porch	Change prior to fourth line after VSYNC
Screen start register 1 Row table mode enable	Change prior to the horizontal blanking interval of the last line of character row before row where new value is to be used

Cursor Address Registers

The contents of these registers define the buffer memory address of the cursor. The cursor output will be asserted when the memory address counter matches the value of the cursor address registers for the scan lines specified in IR6. The cursor address registers can be read or written by the CPU or incremented via the 'increment cursor address' command. In independent buffer mode, these registers define a buffer memory address for AVDC controlled access in response to 'read/write at cursor with/without increment' commands, or the first address to be used in executing the 'write from cursor to pointer' command.

In the independent mode, the RDFLG bit of the status register should be checked for the ready state (bit 5 equal to a logic one) before writing to the cursor address registers. Checking the status register will prevent the cursor address from being changed while a cursor delayed command (e.g., write from cursor to pointer) is still being executed.

Interrupt/Status Registers

The interrupt and status registers provide information to the CPU to allow it to interact with the AVDC to effect desired changes that implement various display operations. The interrupt register provides information on five possible interrupting conditions, as shown in Table 5. These conditions can be selectively enabled or disabled (masked) from causing interrupts by certain AVDC commands. An interrupt condition which is enabled (mask bit equal to one) will cause the $\overline{\text{INTR}}$ output to be asserted and will cause the corresponding bit in the interrupt register to be set-upon the occurrence of the interrupting condition. An interrupt condition which is disabled (mask bit equal to zero) has no effect on either the $\overline{\text{INTR}}$ output or the interrupt register.

The status register provides six bits of status information: the five possible interrupting conditions plus the RDFLG bit. For this register, however, the contents are not affected by the state of the mask bits.

Table 4. Display Control Register Bit Formats

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
COMMAND CODE							
SEE COMMANDS SECTION FOR COMMAND CODES							

COMMAND REGISTERS (WRITE ONLY)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UPPER REGISTER							
Not used		MSBs					

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOWER REGISTER (LSB)							
H'0000' = 0 H'0001' = 1 THRU H'3FFE' = 16,382 H'3FFF' = 16,383 Note: MSBs are in Upper Register [5:0]							

NOTE:
Bits 7 and 6 of upper register are not used in the cursor address register.

CURSOR ADDRESS REGISTERS (READ AND WRITE)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UPPER REGISTER							
DADD15	DADD14	MSBs					

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOWER REGISTER (LSB)							
H'0000' = 0 H'0001' = 1 THRU H'3FFE' = 16,382 H'3FFF' = 16,383 NOTE: MSBs ARE IN UPPER REGISTER [5:0]							

SCREEN START 1 REGISTERS (READ AND WRITE)

NOTES:
1. Bits 7 and 6 of upper register are always zero when read by the CPU.
2. When IR0(7) = 1, the values written into bits 7 and 6 of screen start 1 upper will also be written into IR14(7:6) to control the double width and double height attributes of the display as follows:

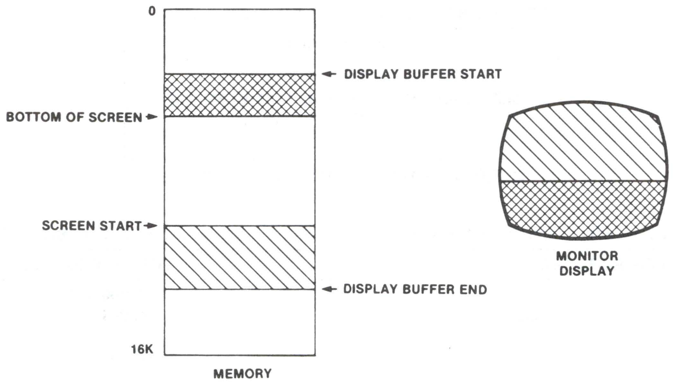
7	6	Attribute
0	0	None
0	1	Double width only
1	0	Double width and double
1	1	Double width and double

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UPPER REGISTER							
SPL2 0 = OFF 1 = ON	SPL1 0 = OFF 1 = ON	MSBs					

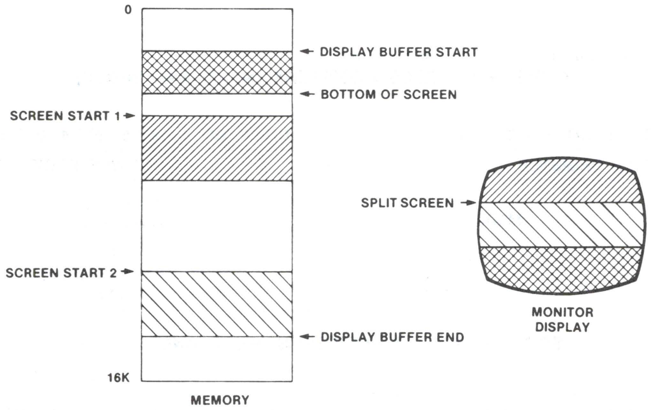
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LOWER REGISTER							
H'0000' = 0 H'0001' = 1 THRU H'3FFE' = 16,382 H'3FFF' = 16,383 NOTE: MSBs ARE IN UPPER REGISTER [5:0]							

SCREEN START 2 REGISTERS (READ AND WRITE)

NOTES:
1. Bit 7 and bit 6 are always zero when read by the CPU.
2. These bits should be set to zero after power-up by the user, even if SSR2 is not used.


0
DISPLAY BUFFER START
BOTTOM OF SCREEN
SCREEN START
DISPLAY BUFFER END
16K
MEMORY
MONITOR DISPLAY

(a) DISPLAY MEMORY WRAPAROUND


0
DISPLAY BUFFER START
BOTTOM OF SCREEN
SCREEN START 1
SPLIT SCREEN
SCREEN START 2
DISPLAY BUFFER END
MONITOR DISPLAY
16K
MEMORY

(b) DISPLAY MEMORY SPLIT SCREEN WITH WRAPAROUND

DF00080S

Figure 17. Display Addressing Operation

Table 5. Interrupt and Status Register Bit Formats

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NOT USED ALWAYS READ AS 0		RDFLG	VBLANK	LINE ZERO	SPLIT 1	READY	SPLIT 2
		0 = BUSY 1 = READY *	0 = NO 1 = YES	0 = NO 1 = YES	0 = NO 1 = YES	0 = BUSY 1 = READY	0 = NO 1 = YES

NOTE:
* Status register only. Always 0 when reading interrupt register.

Descriptions of each interrupt/status register bit follow. Unless otherwise indicated, a bit, once set, will remain set until reset by the CPU by issuing a 'reset interrupt/status bits' command. The bits are also reset by a 'master reset' command and upon power-up.

Sr[5] — RDFLG
This bit is present in the status register only. A zero indicates that the AVDC is currently executing the previously issued delayed command. A one indicates that the AVDC is ready to accept a new delayed command. This bit is set to a one upon a master reset.

I/SR[4] — VBLANK
Indicates the beginning of a vertical blanking interval. Set to one at the beginning of the first scan line of the vertical front porch.

I/SR[3] — Line Zero
Set to one at the beginning of the first scan line (line 0) of each active character row.

I/SR[2] — Split Screen 1
This bit is set when a match occurs between the current character row number and the value contained in split register 1, IR12[6:0]. The equality condition is only checked at the beginning of line zero of each character row.

I/SR[1] — Ready
The delayed commands affect the display and may require the AVDC to wait for a blanking interval before enacting the command. This bit is set to one when execution of a delayed command has been completed. No other delayed command should be invoked until the prior delayed command is completed. This bit is set to a zero upon a master reset.

I/SR[0] — Split Screen 2
This bit is set when a match occurs between the current character row number and the value contained in split register 2 (IR13[6:0]) when you are not scrolling. It is set for the value contained in (split screen register 2) + 1 when scrolling.

COMMANDS

The AVDC commands are divided into two classes: the instantaneous commands which are executed immediately after they are invoked, and the delayed commands which may need to wait for a blanking interval prior to their execution. (Command formats are shown in Table 6). The commands are asserted by performing a write operation to the command register with the appropriate bit pattern as the data byte.

Instantaneous Commands

The instantaneous commands are executed immediately after the trailing edge of the $\overline{WR}$ pulse during which the command is issued. These commands do not affect the state of the RDFLG or READY interrupt/status bits and can be invoked at any time.

Master Reset
This command initializes the AVDC and can be invoked at any time to return the AVDC to its initial state. Upon power-up, two successive master reset commands must be applied to release the AVDC's internal power-on circuits. In transparent and shared buffer modes, the CNTRL1 input must be high when the command is issued. The command causes the following:

1. VSYNC and HSYNC are driven low for the duration of the command and BLANK goes high. After command completion, HSYNC and VSYNC will begin operation and BLANK will remain high until a 'display on' command is received.
2. The interrupt and status bits and masks are set to zero, except for the RDFLG flag which is set to a one.
3. The row buffer mode, cursor off, display off, and the line graphics disable states are set.
4. The initialization register pointer is set to address IR0.
5. IR2[7] is reset.

Load IR Address
This command is used to preset the initialization register pointer with the value 'V' defined by D3 - D0. Allowable values are 0 to 14.

Enable Graphics
After invoking this command, the AVDC will increment the MAC to the next consecutive memory address for each scan line even if more than one scan line per row is programmed. In other words, each scan line begins with a consecutive address from the last displayed address of the previous scan line. This mode can be used for bit-mapped graphics where each location in the display buffer within the defined area contains the bit pattern to be displayed. The graphics mode will be enabled on the next character row after the 'graphics enable' command has been executed. This allows the user to enter and exit graphics mode on character row boundaries.

To perform split screen operations while in graphics mode use SSR2 only.

DADD0/LG is asserted during the trailing edge of BLANK for each scan line while this mode is active.

The AVDC allows up to 128 character rows (initialization register 4) by 256 characters (initialization register 5). For a higher resolution bit mapped screen, the AVDC is programmed as if there are characters and character rows. For example, screen size of 240 × 512 pixels is possible by programming the AVDC for 20 rows with 12 scan lines per row by 64 characters with 8 dots per character.

In the graphics mode, SSR1 should only be updated during the last scan line of the defined 'character row.'

The bit-mapped graphics mode will work only in the independent and transparent modes.

Disable Graphics
Normal addressing resumes at the next row boundary.

Display Off
Asserts the BLANK output. The DADD0 through DADD13 display address bus outputs can be optionally placed in the 3-State condition by setting bit 2 to a '1' when invoking the command.

Display On
Restores normal blanking operation either at the beginning of the next field (bit 2 = 1) or at the beginning of the next scan line (bit 2 = 0). Also returns the DADD0 - DADD13 drivers to their active state.

Cursor Off
Disables cursor operation. Cursor output is placed in the low state.

Cursor On
Enables normal cursor operation.

Reset Interrupt/Status Bits
This command resets the designated bits in the interrupt and status registers. The bit positions correspond to the bit positions in the registers:

Bit 0 — Split 2
Bit 1 — Ready
Bit 2 — Split 1
Bit 3 — Line zero
Bit 4 — Vertical blank

Disable Interrupts
Sets the interrupt mask to zeros for the designated conditions, thus disabling these conditions from being set in the interrupt register and asserting the $\overline{INTR}$ output. Bit position correspondence is as above.

Enable Interrupts
This command writes the associated interrupt mask bits to a one. This enables the corresponding conditions to be set in the interrupt register and asserts the $\overline{INTR}$ output. Bit position correspondence is as above.

Delayed Commands

This group of commands is utilized for the independent buffer mode of operation, although the 'increment cursor' command can also be used in other modes. With the exception of the 'write from cursor to pointer' and 'increment cursor' commands, all the commands of this type will be executed immedi-

Table 6. AVDC Command Formats

D7	D6	D5	D4	D3	D2	D1	D0	Hex	COMMAND
Instantaneous Commands:									
0	0	0	0	0	0	0	0		Master reset
0	0	0	1	V	V	V	V		Load IR pointer with value V (V = 0 to 14)
0	0	1	d	d	d	1	0[1]		Disable graphics
0	0	1	d	d	d	1	1[2]		Enable graphics
0	0	1	d	1	N	d	0[1]		Display off. Float DADD bus if N = 1
0	0	1	d	1	N	d	1[2]		Display on: Next field (N = 1) or scan line (N = 0)
0	0	1	1	d	d	d	0[1]		Cursor off
0	0	1	1	d	d	d	1[2]		Cursor on
0	1	0	N	N	N	N	N		Reset interrupt/status: Bit reset where N = 1
1	0	0	N	N	N	N	N		Disable interrupt: Disable where N = 1
0	1	1	N	N	N	N	N		Enable interrupt: Enables interrupts where N = 1
			V B	L Z	S P 1	R D Y	S P 2		**Interrupt Bit Assignments**
Delayed Commands:								**Hex**	
1	0	1	0	0	1	0	0	A4	Read at pointer address
1	0	1	0	0	0	1	0	A2	Write at pointer address
1	0	1	0	1	0	0	1	A9	Increment cursor address
1	0	1	0	1	1	0	0	AC	Read at cursor address
1	0	1	0	1	0	1	0	AA	Write at cursor address
1	0	1	0	1	1	0	1	AD	Read at cursor address and increment address
1	0	1	0	1	0	1	1	AB	Write at cursor address and increment address
1	0	1	1	1	0	1	1	BB	Write from cursor address to pointer address
1	0	1	1	1	1	0	1	BD	Read from cursor address to pointer address

NOTES:

1. Any combination of these three commands is valid.
2. Any combination of these three commands is valid.
3. d = don't care.
4. No additional circuit required if read latch is implemented using 74LS374.

ately or will be delayed depending on when the command is invoked. If invoked during the active screen time, the command is executed at the next horizontal blanking interval. If invoked during a vertical retrace interval or a 'display off' state, the command is executed immediately.

The 'increment cursor' command is executed immediately after it is issued and requires approximately three $\overline{\text{CCLK}}$ periods for completion. The 'write from cursor to pointer' command executes during blanking intervals. The AVDC will execute as many writes as possible during each blanking interval. If the command is not completed during the current blanking interval, the command will be held in suspension during the next active portion of the screen and continues during the next blanking interval until the command is completed.

In all cases, the AVDC will assert the READY/RDFLG status to signify completion of the delayed command. No other delayed command should be given until the previous delayed command has completed. Therefore, the READY interrupt or RDFLG status flag should be used for handshaking control between the AVDC and CPU when using the delayed commands.

Read/Write at Pointer
Transfers data between the display buffer and the bus interface latch using the address contained in the pointer registers.

Read/Write at Cursor
Transfers data between the display buffer and the bus interface latch using the address contained in the cursor registers.

Increment Cursor
Adds one (modulo 16k) to the cursor address registers. Also note that in place of "Increment cursor" command, "Read/Write at cutsor and increment" command may be used.

Read/Write at Cursor and Increment
Transfers data between the display buffer and the bus interface latch using the address contained in the cursor registers and then adds one (modulo 16k) to the cursor address registers.

Write from Cursor to Pointer
Writes the data contained in the bus interface latch into the block of display memory designated by the cursor address and pointer address registers, inclusive. After completion of the command, the pointer address will be unchanged, but the cursor register contents will be equal to the pointer address.

Read from Cursor to Pointer
Writes the data from the block of display memory designated by the cursor and pointer addresses inclusive into the bus interface latch. This command can be used for a DMA dump of memory into RAM from the cursor location to the pointer location. After completion of the command, the cursor register contents will equal the pointer register contents.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5.0V \pm 5\%$[4, 5, 6]

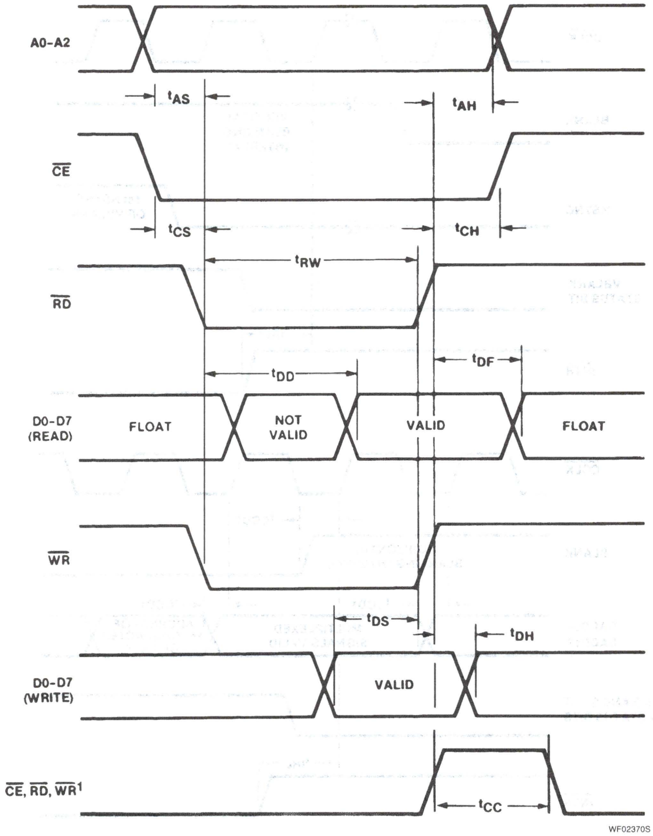
SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except $\overline{INTR}$ output)	$I_{OH} = -200\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	µA
I_{LL}	Data bus 3-State leakage current	$V_O = 0$ to V_{CC}	−10		10	µA
I_{OD}	$\overline{INTR}$ open drain output leakage current	$V_O = 0$ to V_{CC}			10	µA
I_{CC}	Power supply current				185	mA

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%[4, 5, 6, 7]

SYMBOL	PARAMETER	TEST CONDITIONS[8]	2.7MHz		4.0MHz		UNIT
			Min	Max	Min	Max	
Bus timing (Figure 18)[9]							
t_{AS}	A0 – A2 setup time to $\overline{\text{WR}}$, $\overline{\text{RD}}$ low		30		30		ns
t_{AH}	A0 – A2 hold time from $\overline{\text{WR}}$, $\overline{\text{RD}}$ high		0		0		ns
t_{CS}	$\overline{\text{CE}}$ setup time to $\overline{\text{WR}}$, $\overline{\text{RD}}$ low		0		0		ns
t_{CH}	$\overline{\text{CE}}$ hold time from $\overline{\text{WR}}$, $\overline{\text{RD}}$ high		0		0		ns
t_{RW}	$\overline{\text{WR}}$, $\overline{\text{RD}}$ pulse width		250		200		ns
t_{DD}	Data valid after $\overline{\text{RD}}$ low			200		200	ns
t_{DF}	Data bus floating after $\overline{\text{RD}}$ high			100		100	ns
t_{DS}	Data setup time to $\overline{\text{WR}}$ high		150		150		ns
t_{DH}	Data hold time from $\overline{\text{WR}}$ high		10		5		ns
t_{CC}	High time from $\overline{\text{CE}}$ to $\overline{\text{CE}}$						
	Consecutive commands		t_{CCP}		t_{CCP}		ns
	Other accesses		300		300		ns
CCLK timing (Figures 19, 20, 21)							
t_{CCP}	$\overline{\text{CCLK}}$ period		370	10,000	250	10,000	ns
t_{CCH}	$\overline{\text{CCLK}}$ high time		125		100		ns
t_{CCL}	$\overline{\text{CCLK}}$ low time		125		100		ns
	Output delay from $\overline{\text{CCLK}}$ edge[11]						
t_{CCD1}	DADD0 – 13, MBC		40	175	40	150	ns
t_{CCD2}	BLANK, HSYNC, VSYNC/CSYNC, CURSOR, $\overline{\text{BEXT}}$, $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, $\overline{\text{BCE}}$, $\overline{\text{WDB}}$, $\overline{\text{RDB}}$[10]		40	225	40	200	ns
Other timings (Figure 20)							
t_{RDL}	READY/RDFLG low from $\overline{\text{WR}}$ high[9]			t_{CCP} + 30		t_{CCP} + 30	ns
t_{BAK}	$\overline{\text{BACK}}$ high from $\overline{\text{PBREQ}}$ low			225		200	ns
t_{BXT}	$\overline{\text{BEXT}}$ high from $\overline{\text{PBREQ}}$ high			225		200	ns
t_{IRL}	$\overline{\text{INTR}}$ low from $\overline{\text{CCLK}}$ low			225		200	ns
t_{IRH}	$\overline{\text{INTR}}$ high from $\overline{\text{WR}}$, $\overline{\text{RD}}$ high[9]			600		600	ns
t_{AC}	ACLL from HSYNC		3 × t_{CCP}		3 × t_{CCP}		ns
Row table input timing (Figure 21)							
t_{DSRT}	Data setup time to $\overline{\text{CCLK}}$ low		100		60		ns
t_{DHRT}	Data hold time from $\overline{\text{CCLK}}$ low		60		60		ns

NOTES:

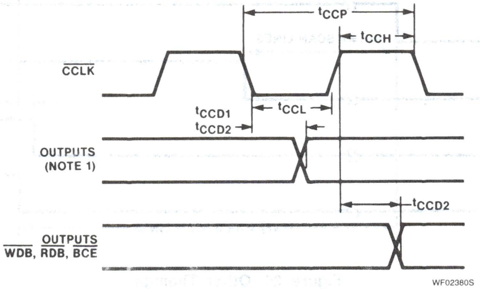
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions above those in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND).
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
8. Test condition for outputs: C_L = 150pF.
9. Timing is illustrated and specified referenced to $\overline{\text{WR}}$ and $\overline{\text{RD}}$ inputs. Device may also be operated with $\overline{\text{CE}}$ as the 'strobing' input. In this case, all timing specifications apply referenced to falling and rising edges of $\overline{\text{CE}}$.
10. $\overline{\text{BCE}}$, $\overline{\text{WDB}}$, and $\overline{\text{RDB}}$ delays track each other within 10ns. Also, these output delays will tend to follow direction (min/max) of DADD0 – 13 delays.
11. These values were measured with a capacitance load of 150pF. To adjust the output delay, use the following correction factor: 50pF $\leqslant C_L <$ 150pF: –0.15ns/pF.


A0-A2
t_{AS}
t_{AH}
$\overline{CE}$
t_{CS}
t_{CH}
t_{RW}
$\overline{RD}$
t_{DF}
t_{DD}
D0-D7 (READ)
FLOAT
NOT VALID
VALID
FLOAT
$\overline{WR}$
t_{DS}
t_{DH}
D0-D7 (WRITE)
VALID
$\overline{CE}$, $\overline{RD}$, $\overline{WR}$[1]
t_{CC}
WF02370S

NOTE:

1. Any two must be high for t_{CC}.

Figure 18. Bus Timing


t_{CCP}
t_{CCH}
$\overline{CCLK}$
t_{CCD1}
t_{CCD2}
t_{CCL}
OUTPUTS (NOTE 1)
t_{CCD2}
OUTPUTS $\overline{WDB}$, $\overline{RDB}$, $\overline{BCE}$
WF02380S

NOTES:

1. DADD0 – DADD13, BLANK, HSYNC, CSYNC/VSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BCE}$, MBC, $\overline{BACK}$.
2. $\overline{BCE}$ changes state on both $\overline{CCLK}$ edges — (see Figures 3 and 4).

Figure 19. $\overline{CCLK}$ Timing

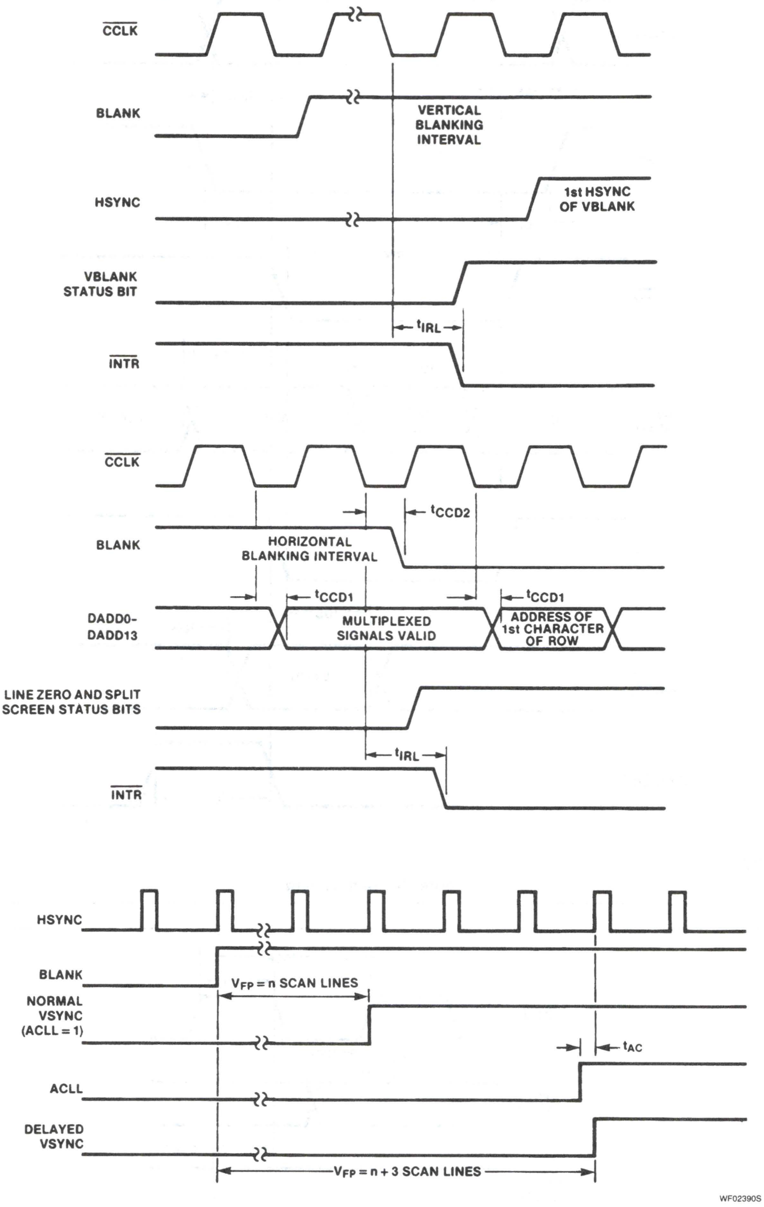

CCLK
BLANK
VERTICAL BLANKING INTERVAL
HSYNC
1st HSYNC OF VBLANK
VBLANK STATUS BIT
t_{IRL}
INTR
CCLK
BLANK
t_{CCD2}
HORIZONTAL BLANKING INTERVAL
t_{CCD1}
DADD0-DADD13
MULTIPLEXED SIGNALS VALID
ADDRESS OF 1st CHARACTER OF ROW
LINE ZERO AND SPLIT SCREEN STATUS BITS
t_{IRL}
INTR
HSYNC
BLANK
V_{FP} = n SCAN LINES
NORMAL VSYNC (ACLL = 1)
t_{AC}
ACLL
DELAYED VSYNC
V_{FP} = n + 3 SCAN LINES
WF02390S

Figure 20. Other Timings

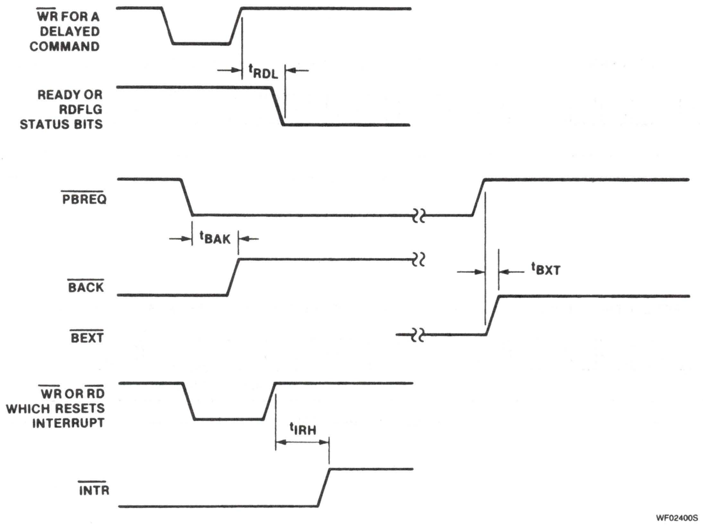

WR FOR A DELAYED COMMAND
tRDL
READY OR RDFLG STATUS BITS
PBREQ
tBAK
tBXT
BACK
BEXT
WR OR RD WHICH RESETS INTERRUPT
tIRH
INTR
WF02400S

Figure 20. Other Timings (Continued)

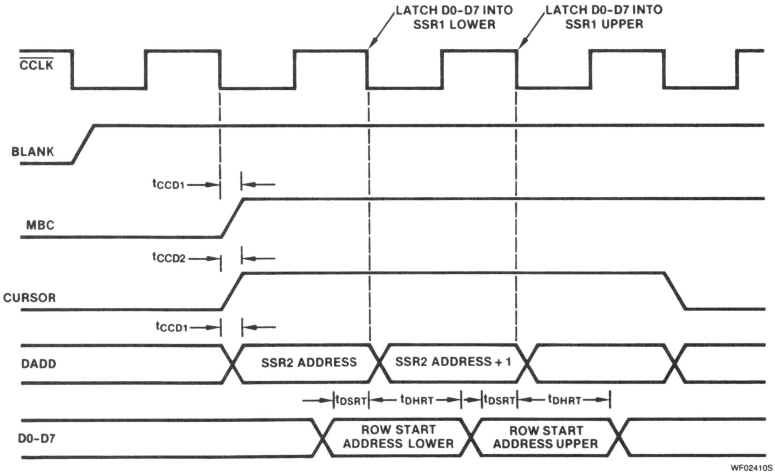
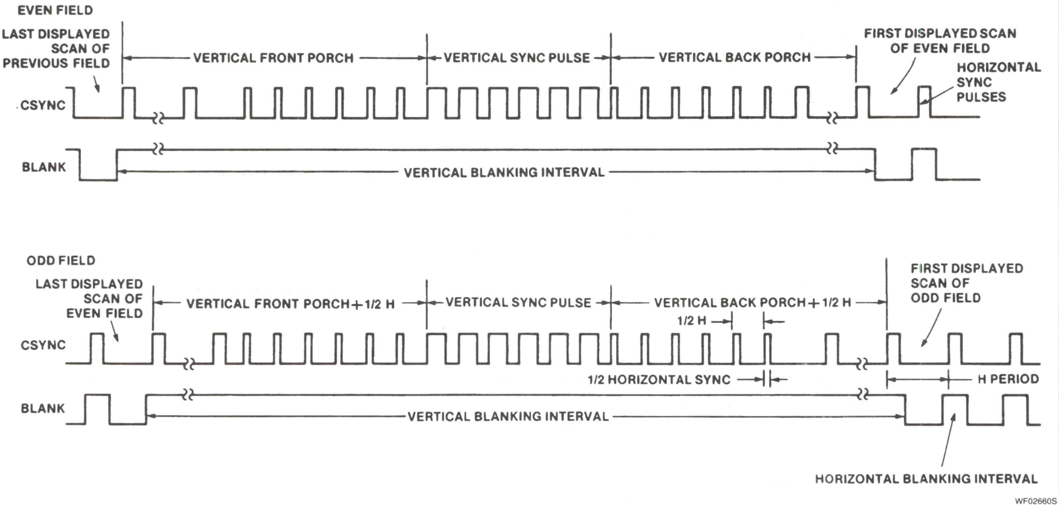

LATCH D0-D7 INTO SSR1 LOWER
LATCH D0-D7 INTO SSR1 UPPER
CCLK
BLANK
tCCD1
MBC
tCCD2
CURSOR
tCCD1
DADD
SSR2 ADDRESS
SSR2 ADDRESS + 1
tDSRT
tDHRT
tDSRT
tDHRT
D0-D7
ROW START ADDRESS LOWER
ROW START ADDRESS UPPER
WF02410S

Figure 21. Row Table Fetch I/O Timing


EVEN FIELD
LAST DISPLAYED SCAN OF PREVIOUS FIELD
VERTICAL FRONT PORCH
VERTICAL SYNC PULSE
VERTICAL BACK PORCH
FIRST DISPLAYED SCAN OF EVEN FIELD
HORIZONTAL SYNC PULSES
CSYNC
BLANK
VERTICAL BLANKING INTERVAL
ODD FIELD
LAST DISPLAYED SCAN OF EVEN FIELD
VERTICAL FRONT PORCH+1/2 H
VERTICAL SYNC PULSE
VERTICAL BACK PORCH+1/2 H
1/2 H
FIRST DISPLAYED SCAN OF ODD FIELD
CSYNC
1/2 HORIZONTAL SYNC
H PERIOD
BLANK
VERTICAL BLANKING INTERVAL
HORIZONTAL BLANKING INTERVAL
WF02660S

NOTES:
1. In non-interlaced operation the even field is repeated continuously.
2. In Interlaced operation the even field alternates with the odd field.

Figure 22. Composite Sync Timing

Signetics

SCB2675
Color/Monochrome Attributes Controller (CMAC)

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics SCB2675 Color/Monochrome Attributes Controller (CMAC) is a bipolar LSI device designed for CRT terminals and display systems that employ raster scan techniques. It contains a programmable dot clock divider to generate a character clock; a high-speed shift register to serialize input dot data into a video stream; latches and logic to apply visual attributes to the resulting display; and logic to display a cursor on the display.

The CMAC provides control of visual attributes on a character-by-character basis for two operating modes: monochrome and color. The monochrome mode provides reverse video, blank, highlight and two general purpose user-definable attributes. In this mode, the display characters can be specified to appear on either a light or dark screen background. Retrace video suppression can be automatically or externally controlled. The color mode provides eight colors for foreground (character) video and eight colors for background video together with a luminance output for external color set selection, or to simultaneously drive a monochrome monitor. Additionally, both modes provide double width, underline, blink, dot stretching and dot width attributes. In monochrome mode, the SCB2675 emulates the attribute characteristics of Digital Equipment Corporation's VT100 terminal.

The horizontal dot frequency is the basic timing input to the CMAC. This clock is divided internally to provide a character clock output for system synchronization. Up to nine bits of dot data are parallel loaded into the video shift register on each character boundary. The two TTL video data outputs in monochrome mode are encoded to provide four video intensities (black, gray, white and highlight). The video data in color mode is encoded to provide eight foreground colors and shifted out on three TTL outputs, together with the luminance output.

FEATURES

- **25MHz and 18MHz video dot rate versions***
- **Four video intensities encoded on two TTL outputs (monochrome mode)**
- **Eight foreground and background colors encoded on three TTL outputs (color mode)**
- **Internally latched character attributes:**
  - **Reverse video**
  - **Blank**
  - **Blink**
  - **Underline**
  - **Highlight**
  - **Two general purpose**
  - **Eight foreground colors**
  - **Eight background colors**
  - **Dot width control**
  - **Double width characters**
- **VT100 compatible attributes**
- **Reverse video cursor with optional white cursor in color mode**
- **Up to 10 dots per character**
- **Light or dark background in monochrome mode**
  - **Automatic retrace blanking**
- **Programmable dot stretching**
- **Compatible with SCN2674 AVDC and SCN2670 DCGG**
- **TTL compatible**
- **40-pin dual-in-line package**

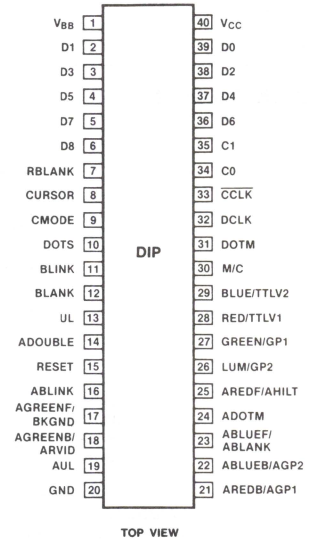
APPLICATIONS

- **CRT terminals**
- **Word processing systems**
- **Small business computers**

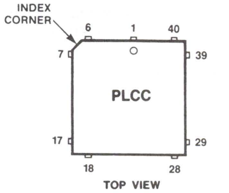
---

*40MHz SCB2675T also available.

PIN CONFIGURATIONS


VBB 1
D1 2
D3 3
D5 4
D7 5
D8 6
RBLANK 7
CURSOR 8
CMODE 9
DOTS 10
BLINK 11
BLANK 12
UL 13
ADOUBLE 14
RESET 15
ABLINK 16
AGREENF/BKGND 17
AGREENB/ARVID 18
AUL 19
GND 20
40 VCC
39 D0
38 D2
37 D4
36 D6
35 C1
34 C0
33 CCLK
32 DCLK
31 DOTM
30 M/C
29 BLUE/TTLV2
28 RED/TTLV1
27 GREEN/GP1
26 LUM/GP2
25 AREDF/AHILT
24 ADOTM
23 ABLUEF/ABLANK
22 ABLUEB/AGP2
21 AREDB/AGP1
DIP
TOP VIEW

CD00430S


INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW

CD0044PS

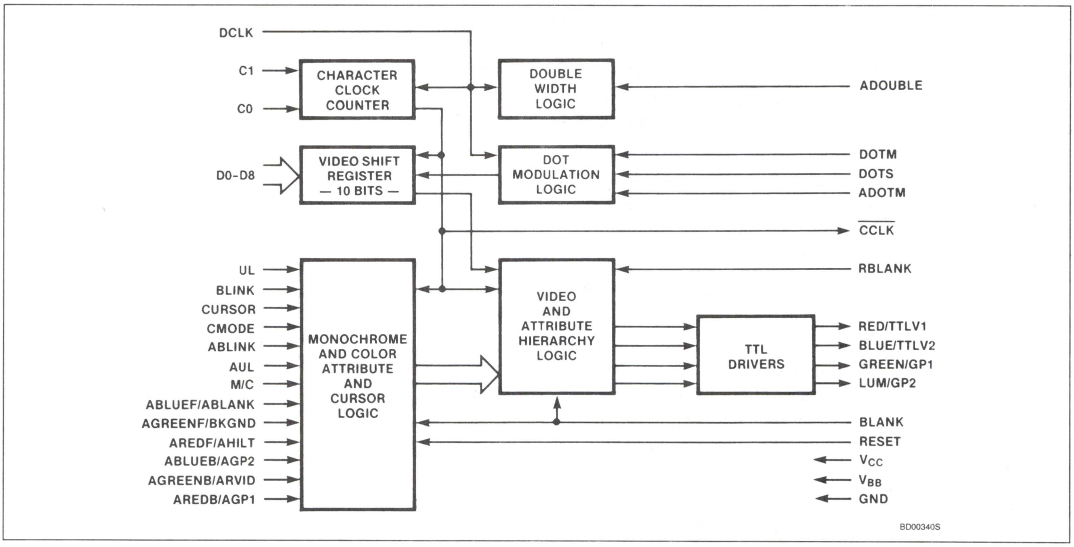
Pin	Function	Pin	Function
2	V_{BB}	24	AREDB/AGP1
3	D1	25	ABLUEB/AGP2
4	D3	26	ABLUEF/ABLANK
5	D5	27	ADOTM
6	D7	28	AREDF/AHILT
7	D8	29	LUM/GP2
8	RBLANK	30	GREEN/GP1
9	CURSOR	31	RED/TTLV1
10	CMODE	32	BLUE/TTLV2
11	DOTS	33	M/C
13	BLINK	35	DOTM
14	BLANK	36	DCLK
15	UL	37	$\overline{\text{CCLK}}$
16	ADOUBLE	38	C0
17	RESET	39	C1
18	ABLINK	40	D6
19	AGREENF/BKGND	41	D4
20	AGREENB/ARVID	42	D2
21	AUL	43	D0
22	GND	44	V_{CC}

Color/Monochrome Attributes Controller (CMAC) SCB2675

ORDERING INFORMATION

PACKAGES	DOTS PER CHARACTER	V_{CC} = +5V ±5%, 0°C to +70°C	
		25MHz	18MHz
Ceramic DIP Plastic DIP Plastic LCC	7, 8, 9, 10	SCB2675BC5I40 SCB2675BC5N40 SCB2675BC5A44	SCB2675BC8I40 SCB2675BC8N40 SCB2675BC8A44
Ceramic DIP Plastic DIP Plastic LCC	6, 8, 9, 10	SCB2675CC5I40 SCB2675CC5N40 SCB2675CC5A44	SCB2675CC8I40 SCB2675CC8N40 SCB2675CC8A44

BLOCK DIAGRAM


DCLK
C1
C0
CHARACTER CLOCK COUNTER
DOUBLE WIDTH LOGIC
ADOUBLE
D0-D8
VIDEO SHIFT REGISTER — 10 BITS —
DOT MODULATION LOGIC
DOTM
DOTS
ADOTM
CCLK
RBLANK
UL
BLINK
CURSOR
CMODE
ABLINK
AUL
M/C
ABLUEF/ABLANK
AGREENF/BKGND
AREDF/AHILT
ABLUEB/AGP2
AGREENB/ARVID
AREDB/AGP1
MONOCHROME AND COLOR ATTRIBUTE AND CURSOR LOGIC
VIDEO AND ATTRIBUTE HIERARCHY LOGIC
TTL DRIVERS
RED/TTLV1
BLUE/TTLV2
GREEN/GP1
LUM/GP2
BLANK
RESET
Vcc
VBB
GND
BD00340S

Color/Monochrome Attributes Controller (CMAC) SCB2675

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
V_{CC}	40	44	I	**Power supply:** +5V ±5%
V_{BB}	1	2	I	**Bias Supply:** See Figure 5
GND	20	22	I	**Ground:** 0V reference
DCLK	32	36	I	**Dot Clock:** Dot frequency input. Video output shift rate.
$\overline{\text{CCLK}}$	33	37	O	**Character Clock:** An output which is a submultiple of DCLK. The period ranges from 6 to 10 DCLK periods per cycle and is determined by the state of the C0, C1 inputs.
RED/TTLV1	28	31	O	**Red/TTL Video 1:** In color mode, this output provides the red gun serial video. In monochrome mode, it should be used with the blue/TTL video 2 output to decode four video intensities.
BLUE/ TTLV2	29	32	O	**Blue/TTL Video 2:** In color mode, this output provides the blue gun serial mode. In monochrome mode, it should be used with the red/TTL video 1 output to decode four video intensities.
GREEN/ GP1	27	30	O	**Green/General Purpose 1:** In color mode, this output provides the green gun serial video. In monochrome mode, it is a general purpose TTL output which is asserted if the AREDB/AGP1 input is asserted when the corresponding character dot data is loaded into the video shift register. GP1 can be active in either active scan or blank time.
LUM/GP2	26	29	O	**Luminance/General Purpose 2:** In color mode, this output is the logical-OR of the RGB foreground video. It is low during a blanking interval and during the foreground portion of the cursor display. In monochrome mode, it is a general purpose TTL output which is asserted if the ABLUEB/AGP2 input is asserted when the corresponding character dot data is loaded into the video shift register. GP2 can be active in either active scan or blank time.
UL	13	15	I	**Underline Timing:** Indicates the scan line(s) for the underline attribute. Latched on the falling edge of BLANK. The underline will be displayed on the specified scan line for every character where AUL = 1.
BLINK	11	13	I	**Blink Timing:** This input is sampled on the falling edge of BLANK to provide the blink rate for the blink attribute. Should be a submultiple of the frame rate.
BLANK	12	14	I	**Screen Blank:** When high, this input forces the video outputs to the specified background color in color mode and to the level specified by the BKGND input (either black or gray) in monochrome mode.
RBLANK	7	8	I	**Retrace Blank:** This input is used to force the video outputs to a low during retrace periods. If pulled high, it will automatically suppress video during the retrace periods when BLANK is high. The user may also pulse this input while BLANK is high to selectively suppress raster video.
AGREENF/ BKGND	17	19	I	**Green Foreground/Background Intensity:** In color mode, this input activates the GREEN/GP1 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input specifies gray or black screen background.
ABLUEF/ ABLANK	23	26	I	**Blue Foreground/Blank Attribute:** In color mode, this input activates the BLUE/TTLV2 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input generates a blank space for the associated character. The blank space intensity is controlled by the AGREENF/BKGND input, the reverse video attribute and cursor input.
AREDF/ AHILT	25	28	I	**Red Foreground/Highlight Attribute:** In color mode, this input activates the RED/TTLV1 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input highlights the associated character (including underline).
CURSOR	8	9	I	**Cursor Timing:** This input provides the timing for the cursor video. In color mode, with CURSOR and CMODE high, the RGB outputs are driven high (white cursor). If CMODE is low, or in monochrome mode, this input reverses the intensities of the video and attributes (the foreground and background intensities are reversed). Cursor position, shape, and blink rate are controlled by this input.
CMODE	9	10	I	**Cursor Mode:** Used in color mode only. When CURSOR and CMODE are high, the RGB outputs are driven high (white cursor). When CURSOR is high and CMODE is low, the RGB outputs are logically inverted (reverse video cursor).
AUL	19	21	I	**Underline Attribute:** Specifies a line to be displayed in the character block. The specific line(s) are specified by the UL input. All other attributes apply to the underline video.
ABLINK	16	18	I	**Blink Attribute:** In color mode, this active high input will drive the foreground RGB combination to the background RGB combination. In monochrome mode, the associated character or background is driven to the intensity determined by BKGND, reverse video attribute and the cursor input.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
ADOUBLE	14	16	I	**Double Width Attribute:** This active high input causes the associated character video to be shifted out of the serial shift register at one half the dot frequency (DCLK). The $\overline{CCLK}$ output is not affected.
AREDB/ AGP1	21	24	I	**Red Background/General Purpose Attribute 1:** In color mode, this input activates the RED/ TTLV1 output during the background portion of the associated character block. In monochrome mode, it activates the GREEN/GP1 output for the associated character block.
ABLUEB/ AGP2	22	25	I	**Blue Background/General Purpose Attribute 2:** In color mode, this input activates the BLUE/ TTLV2 output during the background portion of the associated character block. In monochrome mode, it activates the LUM/GP2 output for the associated character block.
AGREENB/ ARVID	18	20	I	**Green Background/Reverse Video Attribute:** In color mode, this input activates the GREEN/ GP1 output during the background portion of the associated character block. In monochrome mode, it causes the associated character block video intensities to be reversed.
D0 – D8	39, 2, 38, 3, 37, 4, 36, 5, 6	43, 3, 42, 4, 41, 5, 40, 6, 7	I	**Dot Data Input:** These are parallel inputs corresponding to the character/graphic symbol dot data for a given scan line. These inputs are strobed into the video shift register on the trailing (falling) edge of each character clock ($\overline{CCLK}$).
C0, C1	34, 35	38, 39	I	**Character Clock Control:** The states of these two static inputs determine the internal divide factor for the $\overline{CCLK}$ output rate.
RESET	15	17	I	**Reset:** This active high input initializes the internal logic and resets the attribute latches.
M/C	30	33	I	**Monochrome/Color Mode:** This input selects whether the CMAC operates in monochrome or color mode. A low selects color mode and a high selects monochrome mode.
ADOTM	24	27	I	**Dot Modulation Attribute:** When DOTM and this input are high, the active dot width of the associated character video is one DCLK. When DOTM is high and this input is low, the active dot width of the associated character video is two DCLKs.
DOTM	31	35	I	**Dot Width Modulation:** When this input is high, two DCLKs are used for each dot shifted through the shift register. When this input is low, one DCLK is used.
DOTS	10	11	I	**Dot Stretching:** Sampled at the falling edge of BLANK. When this input is high, one extra dot is appended to individual dots or groups of dots of the input parallel data and then transferred through the shift register. When this input is low, normal transfer of input parallel data results.

FUNCTIONAL DESCRIPTION

The CMAC consists of seven major sections (see block diagram). The high-speed dot clock input is applied to a programmable divider to provide a character clock output for system timing. Parallel dot data is loaded into the video shift register on character boundaries and shifted into the video logic block at the dot rate specified by the dot modulation section. The appropriate attribute control inputs are selected by the mode select logic, latched internally on character boundaries, and combined with the serial dot data to provide monochrome or color video outputs.

The BLANK input defines the active screen and retrace areas. In color mode, the video outputs are forced to the specified background color when this signal is asserted; in monochrome mode the video outputs are forced to the states defined by the BKGND input, i.e., black if dark background is selected and gray if light background is selected. A separate RBLANK input allows the user to select the amount of border around the active area when operating in color mode or in monochrome mode with light background.

Table 1. Monochrome Mode Attribute Characteristics

REV[1]	AHILT	ABLINK[2]	FOREGROUND VIDEO	BACKGROUND VIDEO
0	0	0	W	B
0	0	1	W/G	B
0	1	0	H	B
0	1	1	H/W	B
1	0	0	B	G
1	0	1	B/W	G/B
1	1	0	B	W
1	1	1	B/H	W/B

NOTES:

1. REV = (BKGND) XOR (ARVID):

BKGND	ARVID	REV
0	0	0
0	1	1
1	0	1
1	1	0

2. For blinking, the video outputs are shown as 0/1, where 0 and 1 are the blink timing input states.
3. Foreground includes underline when underlining is specified by AUL = 1.
4. When ABLANK = 1, foreground component becomes same as background component.
5. Codes for video outputs are as follows:

CODE	TTLV2	TTLV1	BEAM INTENSITY
B	0	0	Black (B)
G	0	1	Gray (G)
W	1	0	White (W)
H	1	1	Highlight (H)

This input can be tied high, in which case the area outside the active area will be dark, or it may be pulsed during BLANK periods to externally control the border widths.

In color mode, eight colors for the character (foreground) and eight colors for the background (area other than character) can be selected by the attribute inputs. In monochrome mode, the intensities of foreground and background are a function of the attribute and BKGND inputs, i.e., characters may be black, gray, white, or highlight (very white) while background may be black, gray, or white (see Table 1).

Character Clock Counter

The character clock counter divides the DCLK input to generate the character clock ($\overline{\text{CCLK}}$). The divide factor is specified by the clock control inputs (C1, C0) as follows:

C1	C0	SCB2675B	
		Dots/ Char.	$\overline{\text{CCLK}}$ Duty Cycle*
0	0	10	5/5
0	1	7	3/4
1	0	8	4/4
1	1	9	4/5

NOTE:
*High/low

C1	C0	SCB2675C	
		Dots/ Char.	$\overline{\text{CCLK}}$ Duty Cycle*
0	0	10	5/5
0	1	6	3/3
1	0	8	4/4
1	1	9	4/5

NOTE:
*High/low

The number of dot clocks/character is normally the number of dots/character as listed above. However, when dot width control is specified, the DCLK input is divided by two before it is applied to the character clock counter resulting in the number of dot clocks/character being double those listed above, although the number of displayed dots/character remains the same. The number of dots per character (C1, C0) can be dynamically changed on either edge of the character clock as shown in Figure 10 (See Dot Modulation Logic).

Video Shift Register

On each character boundary, the parallel input dot data (D0 – D8) is loaded into the video shift register. The data is shifted out least significant bit first (D0) at the DCLK rate. If 10 dots/character are specified (C1, C0 = 00), the tenth dot will be the same as D8. The serial dot data from the video shift register is routed to the video logic where it is combined with the cursor and attribute control bits to produce the video data outputs.

Mode Select, Attribute And Cursor Control

The mode select logic multiplexes the monochrome and color attribute inputs and outputs as specified by the M/C input. The monochrome mode provides blank, reverse video, highlight and two general purpose attributes. The latter may be used, with external logic, to combine other attributes (e.g., overscore) into the video stream. The color mode provides RGB foreground and background color attributes. Both modes provide double width characters, blink, underline, dot width control and dot stretching.

The cursor and attribute inputs are pipelined internally to allow for system pipeline propagations. The cursor input and the attribute inputs are delayed for one $\overline{\text{CCLK}}$ to account for the delay of the character data through the character generator latches. The attribute timing inputs (BLINK, UL and DOTS) are clocked into the 2675 at the beginning of each scan line time by the falling edge of BLANK. Thus, these inputs must be in their proper state at the falling edge of BLANK preceding the scan line where they are required to be active. The BLANK signal itself is also delayed internally to provide for the RAM and character generator delays (see Figures 6 and 7). Internal delays cause the video outputs to be delayed relative to $\overline{\text{CCLK}}$ as illustrated in Figure 8.

Video Logic

Each character block consists of the three components shown in Figure 1. Symbol video is generated from the dot data inputs D0 – D8. Underline video is enabled by the AUL attribute and is generated during the scan lines for which the UL input is active. Underline and symbol video are always the same intensity or color, and other attributes (e.g., ABLINK) apply to them equally. The combination of underline and symbol video is also referred to as foreground video. Background video is the area of the character block corresponding to the absence of foreground video. The assertion of the non-display attribute in the monochrome mode (ABLANK) causes the entire character block to be displayed as background.

In monochrome mode, the serial dot data and pipelined cursor and attributes are combined to generate four video intensities (black, gray, white and highlight) which are encoded on the TTLV1 and TTLV2 outputs as follows:

TTLV2	TTLV1	VIDEO INTENSITY
0	0	Black
0	1	Gray
1	0	White
1	1	Highlight

Table 1 describes the relationship between attributes and video intensity of the foreground and background components of the character block in monochrome mode.

In color mode, the colors of the foreground and background components are specified by the corresponding attribute inputs; AREDF, AGREENF and ABLUEF dictate the color of the foreground component while AREDB, AGREENB and ABLUEB do the same for the background component. In this mode, the serial dot data and pipelined cursor and attributes are combined to generate four video outputs. The RED, GREEN and BLUE outputs separately contain the corresponding foreground and background components. The LUM output is the logical-OR of the foreground colors and can be used to drive a separate monochrome monitor or to select a different set of colors for the foreground.

Dot Modulation Logic

The dot modulation logic controls the video shift register to supply dot stretching and dot width control.

Dot stretching is controlled by the DOTS input which is sampled by each scan line at the trailing (falling) edge of BLANK. If DOTS is asserted at that time, all characters on the following scan line will have dot stretching applied. Dot stretching causes an extra dot to be added to individual dots or groups of dots as shown in Figures 2 and 3. Dot stretching also can be used to:

1. Compensate for low video bandwidth monitors (since the minimum active displayed segment with dot stretching is two DCLKs).
2. Assure crisp black characters when operating in white background mode.
3. Provide thick characters as a means of distinguishing areas of the display.

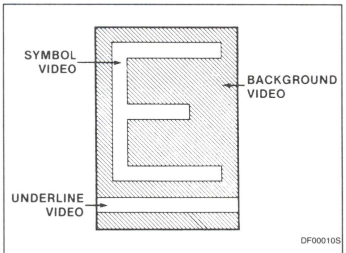


Figure 1. Character Block Definition

Color/Monochrome Attributes Controller (CMAC) SCB2675

Dot width is controlled by the DOTM and ADOTM inputs. DOTM is tied either high, which enables the feature on the entire display, or low, which disables the feature. With ADOTM high, the dot width of characters can be selectively controlled by assertion of the ADOTM attribute input. When operating in this mode, the dot clock input is divided by two before being applied to other circuits in the CMAC. The $\overline{\text{CCLK}}$ output is also divided by two.

The truth table for the possible combinations of ADOTM and DOTM is as follows:

ADOTM	DOTM	OPERATION
0	0	Normal mode
0	1	Dot width control, 100% duty cycle (DCLK÷2, CCLK÷2)
1	0	Not allowed
1	1	Dot width control, 50% duty cycle (DCLK÷2, CCLK÷2)

When dot width control is enabled as above, two DCLKs are used for each video dot period. Asserting ADOTM for a particular character will cause each active video dot of the displayed character to be turned on for one DCLK and off for the other DCLK, while if ADOTM is negated for that character, the active video dot for that character will be turned on (black background) or off (white background) for both DCLK times (see Figures 2 and 4). Only the character video component of the character block is modulated. Underline video and background are not affected by ontime modulation. Width control can be used to:

1. Make horizontal lines and vertical lines appear the same brightness on the display.
2. Provide two different brightness levels for characters without requiring a monitor with analog brightness inputs.

However, note that the effects produced by this feature are highly dependent on the video amplifier characteristics of the monitor used.

Double Width Logic

The double width logic controls the rate at which dots are shifted through the video shift register. When the ADOUBLE input is asserted, the associated character video will be shifted at one half the DCLK rate, and the dot information for the next character will be loaded into the shift register two $\overline{\text{CCLK}}$s later. The character and attribute data that is present on the first CCLK after the ADOUBLE input is asserted will be ignored. The $\overline{\text{CCLK}}$ output is not affected. If a double width character is specified at the last location of a character row, the second half of the double width character (one $\overline{\text{CCLK}}$) will extend into the horizontal front porch.

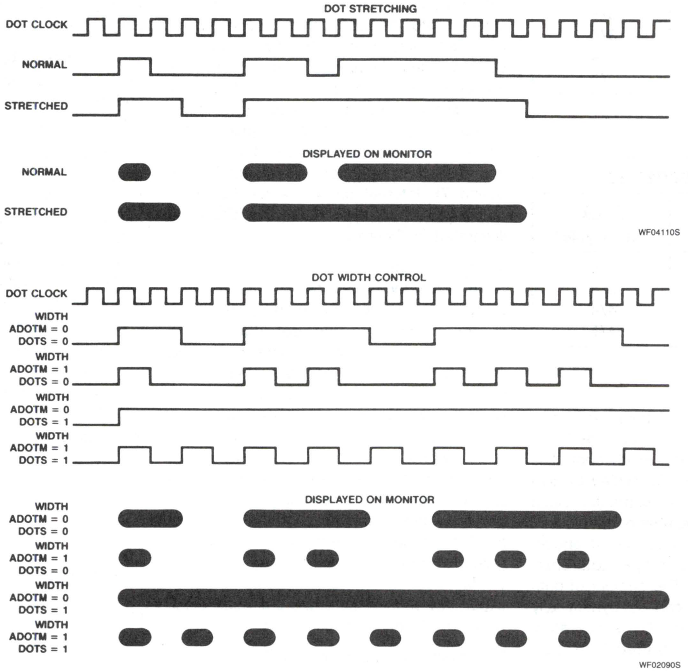

DOT STRETCHING
DOT CLOCK
NORMAL
STRETCHED
DISPLAYED ON MONITOR
NORMAL
STRETCHED
WF04110S
DOT WIDTH CONTROL
DOT CLOCK
WIDTH ADOTM = 0 DOTS = 0
WIDTH ADOTM = 1 DOTS = 0
WIDTH ADOTM = 0 DOTS = 1
WIDTH ADOTM = 1 DOTS = 1
DISPLAYED ON MONITOR
WIDTH ADOTM = 0 DOTS = 0
WIDTH ADOTM = 1 DOTS = 0
WIDTH ADOTM = 0 DOTS = 1
WIDTH ADOTM = 1 DOTS = 1
WF02090S

Figure 2. Dot Modulation Timing

Color/Monochrome Attributes Controller (CMAC) SCB2675

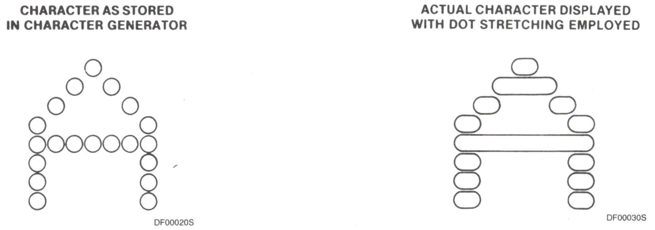

CHARACTER AS STORED
IN CHARACTER GENERATOR
ACTUAL CHARACTER DISPLAYED
WITH DOT STRETCHING EMPLOYED
DF00020S
DF00030S

Figure 3. Dot Stretching

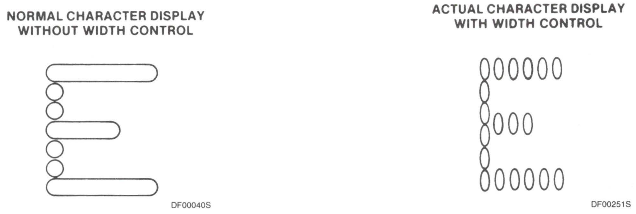

NORMAL CHARACTER DISPLAY
WITHOUT WIDTH CONTROL
ACTUAL CHARACTER DISPLAY
WITH WIDTH CONTROL
DF00040S
DF00251S

Figure 4. Dot Width Control

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$, V_{BB} = Figure 5[4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2			V
V_{OL}	Output low voltage	$I_{OL} = 4mA$			0.4	V
V_{OH}	Output high voltage	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input low current	$V_{IN} = 0.4V$				
	DCLK				−800	μA
	All other inputs				−400	μA
I_{IH}	Input high current	$V_{IN} = 2.4V$				
	DCLK				40	μA
	All other inputs				20	μA
I_{CC}	V_{CC} supply current	$V_{IN} = 0V$, V_{CC} = Max			80	mA
I_{BB}	V_{BB} supply current	Figure 5			120	mA

Color/Monochrome Attributes Controller (CMAC) SCB2675

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$, V_{BB} = Figure 5[4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS				UNIT
			25MHz Version		18MHz Version		
			Min	Max	Min	Max	
Dot clock timing (see Figure 8)							
f_D	Frequency			25		18	MHz
t_{DH}	High time		15		22		ns
t_{DL}	Low time		15		22		ns
Setup times (see Figures 6, 7, 9, and 10)							
t_{SB}	BLANK to $\overline{\text{CCLK}}$		40		50		ns
t_{SA}	Attributes to $\overline{\text{CCLK}}$		40		50		ns
t_{SD}	D0 – D9 to $\overline{\text{CCLK}}$		70		70		ns
t_{SK}	CURSOR to $\overline{\text{CCLK}}$		40		50		ns
t_{SC}	C0, C1 to DCLK		30		35		ns
t_{SR}	RBLANK to DCLK		20		20		ns
t_{SM}	BLINK, UL, DOTS to BLANK		20		20		ns
Hold times (see Figures 6, 7, 9, and 10)							
t_{HB}	BLANK from $\overline{\text{CCLK}}$		20		20		ns
t_{HA}	Attributes from $\overline{\text{CCLK}}$		20		20		ns
t_{HD}	D0 – D8 from $\overline{\text{CCLK}}$		30		30		ns
t_{HK}	CURSOR from $\overline{\text{CCLK}}$		20		20		ns
t_{HC}	C0, C1 from DCLK		20		20		ns
t_{HR}	RBLANK from DCLK		20		20		ns
t_{HM}	BLINK, UL, DOTS from BLANK		20		20		ns
Delay times (see Figure 8)							
t_{DC}	$\overline{\text{CCLK}}$ from DCLK	$C_L = 50pF$		55		70	ns
t_{DV}	Other outputs from DCLK			60		70	ns

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. For testing, all input signals swing between 0.4V and 2.4V with a transition time of 3ns maximum and output voltages are checked at 0.8V and 2V.

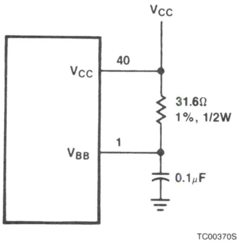
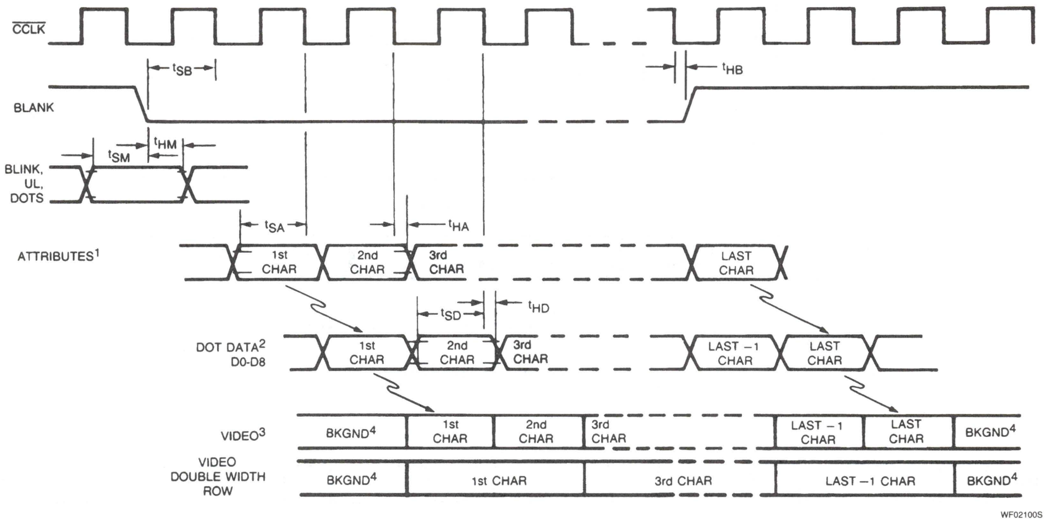


Figure 5. Recommended V_{BB} Test Circuit


CCLK
t_{SB}
t_{HB}
BLANK
t_{SM}
t_{HM}
BLINK, UL, DOTS
t_{SA}
t_{HA}
ATTRIBUTES1
1st CHAR
2nd CHAR
3rd CHAR
LAST CHAR
t_{SD}
t_{HD}
DOT DATA2 D0-D8
1st CHAR
2nd CHAR
3rd CHAR
LAST −1 CHAR
LAST CHAR
VIDEO3
BKGND4
1st CHAR
2nd CHAR
3rd CHAR
LAST −1 CHAR
LAST CHAR
BKGND4
VIDEO DOUBLE WIDTH ROW
BKGND4
1st CHAR
3rd CHAR
LAST −1 CHAR
BKGND4
WF02100S

NOTES:

1. Attributes include: ABLINK, ABLANK, ARVID, AUL, AHILT, ADOUBLE, ADOTM, two general purpose, and foreground/background colors.
2. One CCLK delay for dot data (obtained from delay through character generator).
3. See Figure 8 for detail timing of video outputs.
4. Non-active scan time. VIDEO reverts to polarity selected by the BKGND input in monochrome mode.

Figure 6. CMAC Pipeline Timing

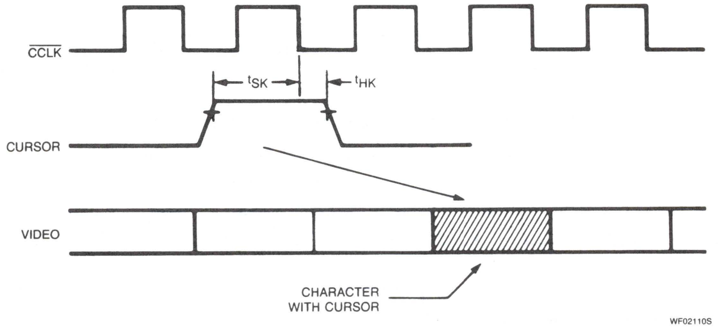

CCLK
t_{SK}
t_{HK}
CURSOR
VIDEO
CHARACTER WITH CURSOR
WF02110S

Figure 7. Cursor Pipeline Timing

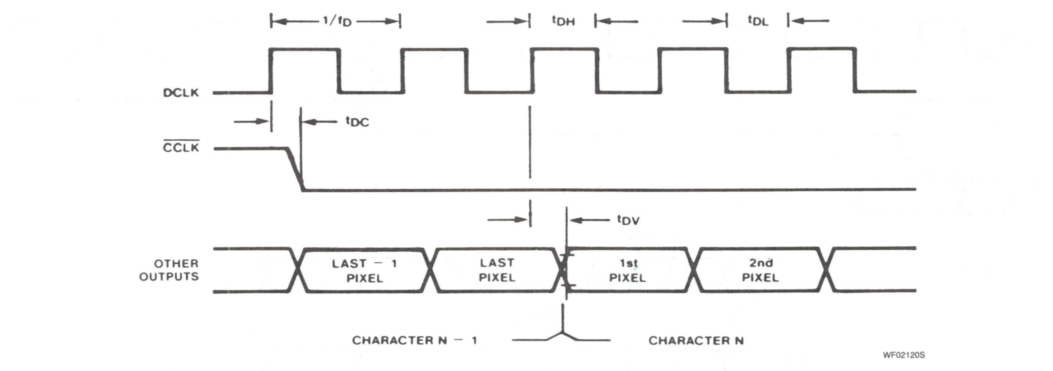

1/fD
tDH
tDL
DCLK
tDC
CCLK
tDV
OTHER OUTPUTS
LAST − 1 PIXEL
LAST PIXEL
1st PIXEL
2nd PIXEL
CHARACTER N − 1
CHARACTER N
WF02120S

Figure 8. Output Pipeline Timing

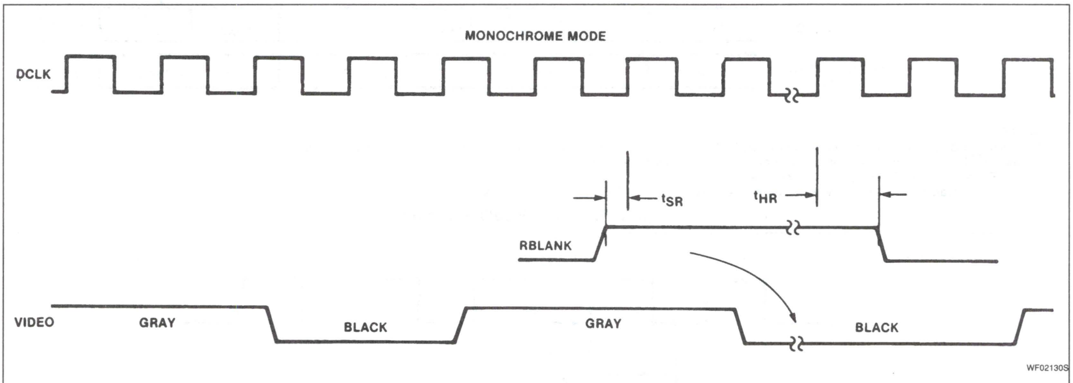
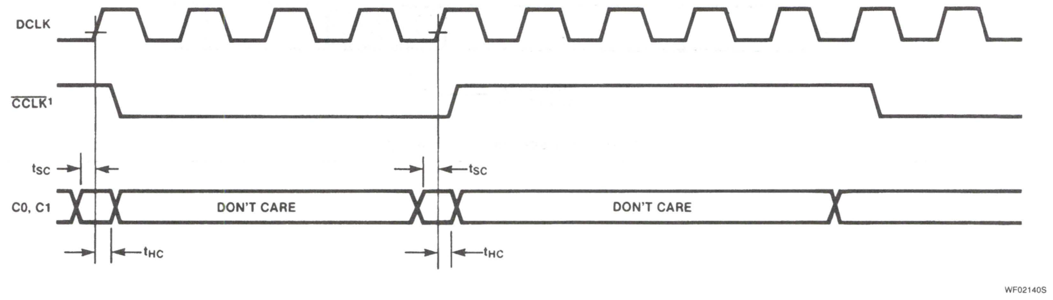

MONOCHROME MODE
DCLK
tSR
tHR
RBLANK
VIDEO
GRAY
BLACK
GRAY
BLACK
WF02130S

Figure 9. RBLANK Timing During Inactive Scan Time (Blank = 1)


DCLK
CCLK¹
tSC
tSC
C0, C1
DON'T CARE
DON'T CARE
tHC
tHC
WF02140S

NOTE:
1. The high and low times of CCLK may be controlled independently.

Figure 10. Clock Divider Timing

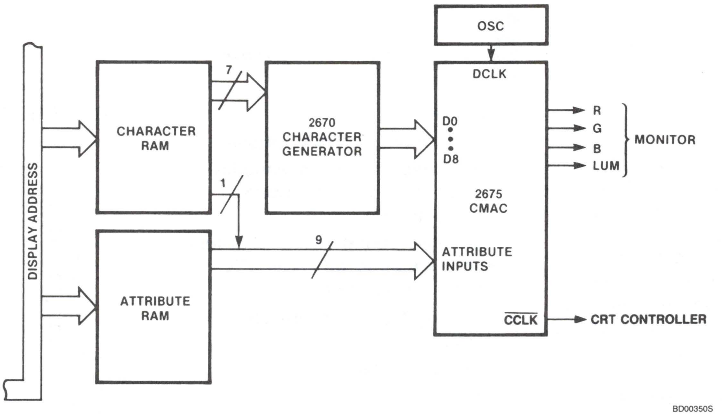

OSC
DCLK
DISPLAY ADDRESS
CHARACTER RAM
7
2670 CHARACTER GENERATOR
D0
D8
R
G
B
LUM
MONITOR
1
2675 CMAC
9
ATTRIBUTE INPUTS
ATTRIBUTE RAM
CCLK
CRT CONTROLLER
BD00350S

Figure 11. System Block Diagram of SCB2675 in Color Mode

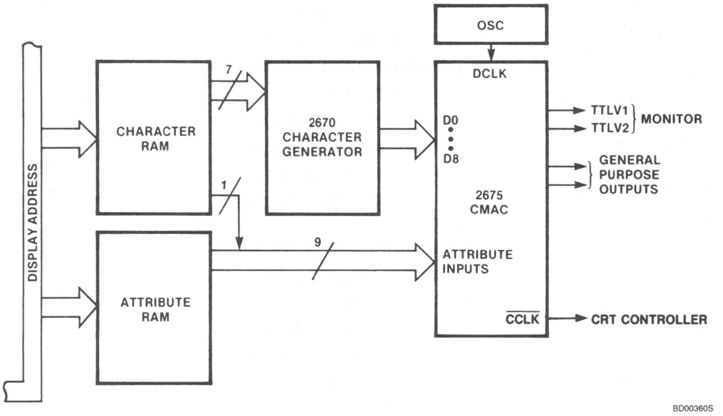

OSC
DCLK
DISPLAY ADDRESS
CHARACTER RAM
7
2670 CHARACTER GENERATOR
D0
D8
TTLV1
TTLV2
MONITOR
GENERAL PURPOSE OUTPUTS
1
2675 CMAC
9
ATTRIBUTE INPUTS
ATTRIBUTE RAM
CCLK
CRT CONTROLLER
BD00360S

Figure 12. System Block Diagram of SCB2675 in Monochrome Mode

Signetics

Microprocessor Products

SCB2675T

Turbo Color/Monochrome Attributes Controller (Turbo-CMAC)

Preliminary Specification

DESCRIPTION

The Signetics SCB2675T Turbo Color/Monochrome Attr ibutes Controller (Turbo-CMAC) is a bipolar LSI device designed for CRT terminals and display systems that employ raster scan techniques. It contains a programmable dot clock divider to generate a character clock, a high-speed shift register to serialize input dot data into a video stream, latches and logic to apply visual attributes to the resulting display, and logic to display a cursor on the display.

The Turbo-CMAC provides control of visual attributes on a character by character basis for two operating modes: monochrome and color. The monochrome mode provides reverse video, blank, highlight and two general purpose user definable attributes. In this mode, the display characters can be specified to appear on either a light or dark screen background. Retrace video suppression can be automatically or externally controlled. The color mode provides eight colors for foreground (character) video and eight colors for background video together with a luminance output for external color set selection or to simultaneously drive a monochrome monitor. Additionally, both modes provide double width, underline, blink, dot stretching and dot width attributes. In monochrome mode, the SCB2675T emulates the attribute characteristics of Digital Equipment Corporation's VT100 terminal.

The horizontal dot frequency is the basic timing input to the Turbo-CMAC. This clock is divided internally to provide a character clock output for system synchronization. Up to nine bits of dot data are parallel loaded into the video shift register on each character boundary. The two TTL video data outputs in monochrome mode are encoded to provide four video intensities (black, gray, white and highlight). The video data in color mode is encoded to provide eight foreground colors and shifted out on three TTL outputs, together with the luminance output.

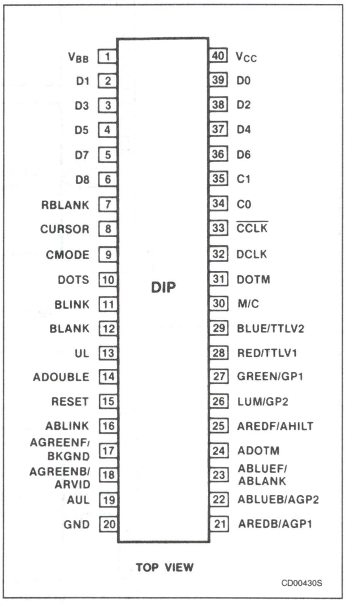
FEATURES

- **40MHz video dot rate version**
- **Four video intensities encoded on two TTL outputs (monochrome mode)**
- **Eight foreground and background colors encoded on three TTL outputs (color mode)**
- **Internally latched character attributes:**
  - **Reverse video**
  - **Blank**
  - **Blink**
  - **Underline**
  - **Highlight**
  - **Two general purpose**
  - **Eight foreground colors**
  - **Eight background colors**
  - **Dot width control**
  - **Double width characters**
- **VT100 compatible attributes**
- **Reverse video cursor with optional white cursor in color mode**
- **Up to 10 dots per character**
- **Light or dark background in monochrome mode**
  - **Automatic retrace blanking**
- **Programmable dot stretching**
- **Compatible with SCN2674 AVDC and SCN2670 DCGG**
- **TTL compatible**
- **40-pin dual in-line package**

APPLICATIONS

- **CRT terminals**
- **Word processing systems**
- **Small business computers**

PIN CONFIGURATION

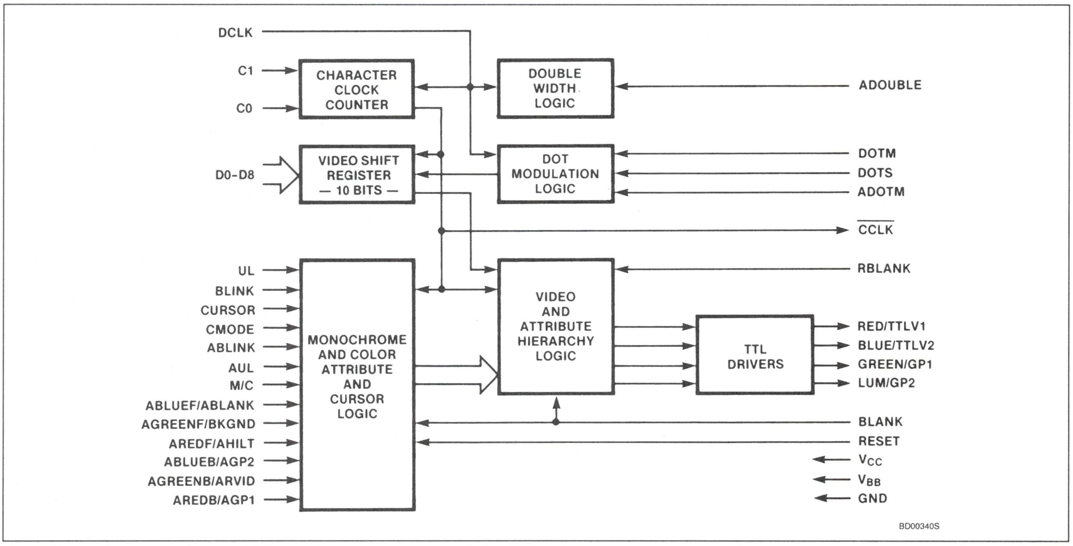

V_{BB} 1
D1 2
D3 3
D5 4
D7 5
D8 6
RBLANK 7
CURSOR 8
CMODE 9
DOTS 10
BLINK 11
BLANK 12
UL 13
ADOUBLE 14
RESET 15
ABLINK 16
AGREENF/BKGND 17
AGREENB/ARVID 18
AUL 19
GND 20
40 V_{CC}
39 D0
38 D2
37 D4
36 D6
35 C1
34 C0
33 $\overline{CCLK}$
32 DCLK
31 DOTM
30 M/C
29 BLUE/TTLV2
28 RED/TTLV1
27 GREEN/GP1
26 LUM/GP2
25 AREDF/AHILT
24 ADOTM
23 ABLUEF/ABLANK
22 ABLUEB/AGP2
21 AREDB/AGP1
DIP
TOP VIEW
CD00430S

Turbo Color/Monochrome Attributes Controller (Turbo-CMAC) SCB2675T

ORDERING INFORMATION

PACKAGES	DOTS PER CHARACTER	V_{CC} = 5V ±5%, 0°C to +70°C 40MHz
Plastic DIP	7, 8, 9, 10	SCB2675TC4N40

BLOCK DIAGRAM


DCLK
C1
C0
CHARACTER CLOCK COUNTER
DOUBLE WIDTH LOGIC
ADOUBLE
D0-D8
VIDEO SHIFT REGISTER — 10 BITS —
DOT MODULATION LOGIC
DOTM
DOTS
ADOTM
CCLK
RBLANK
UL
BLINK
CURSOR
CMODE
ABLINK
AUL
M/C
ABLUEF/ABLANK
AGREENF/BKGND
AREDF/AHILT
ABLUEB/AGP2
AGREENB/ARVID
AREDB/AGP1
MONOCHROME AND COLOR ATTRIBUTE AND CURSOR LOGIC
VIDEO AND ATTRIBUTE HIERARCHY LOGIC
TTL DRIVERS
RED/TTLV1
BLUE/TTLV2
GREEN/GP1
LUM/GP2
BLANK
RESET
V_{CC}
V_{BB}
GND
BD00340S

Turbo Color/Monochrome Attributes Controller (Turbo-CMAC) SCB2675T

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
	DIP		
V_{CC}	40	I	**Power Supply:** $+5V_{DC}$
V_{BB}	1	I	**Bias Supply:** See Figure 5
GND	20	I	**Ground:** 0V reference
DCLK	32	I	**Dot Clock:** Dot frequency input. Video output shift rate.
$\overline{\text{CCLK}}$	33	O	**Character Clock:** An output which is a submultiple of DCLK. The period ranges from 7 to 10 DCLK periods per cycle and is determined by the state of the C0-C1 inputs.
RED/TTLV1	28	O	**Red/TTL Video 1:** In color mode, this output provides the red gun serial video. In monochrome mode, it should be used with the blue/TTL video 2 output to decode four video intensities.
BLUE/TTLV2	29	O	**Blue/TTL Video 2:** In color mode, this output provides the blue gun serial video. In monochrome mode, it should be used with the red/TTL video 1 output to decode four video intensities.
GREEN/GP1	27	O	**Green/General Purpose 1:** In color mode, this output provides the green gun serial video. In monochrome mode, it is a general purpose TTL output which is asserted if the AREDB/AGP1 input is asserted when the corresponding character dot data is loaded into the video shift register. GP1 can be active in either active scan or blank time.
LUM/GP2	26	O	**Luminance/General Purpose 2:** In color mode, this output is the logical-OR of the RGB foreground video. It is low during a blanking interval and during the foreground portion of the cursor display. In monochrome mode, it is a general purpose TTL output which is asserted if the ABLUEB/AGP2 input is asserted when the corresponding character dot data is loaded into the video shift register. GP2 can be active in either active scan or blank time.
UL	13	I	**Underline Timing:** Indicates the scan line(s) for the underline attribute. Latched on the falling edge of BLANK. The underline will be displayed on the specified scan line for every character where AUL = 1.
BLINK	11	I	**Blink Timing:** This input is sampled on the falling edge of BLANK to provide the blink rate for the blink attribute. Should be a submultiple of the frame rate.
BLANK	12	I	**Screen Blank:** When high, this input forces the video outputs to the specified background color in color mode and to the level specified by the BKGND input (either black or gray) in monochrome mode.
RBLANK	7	I	**Retrace Blank:** This input is used to force the video outputs to a low during retrace periods. If pulled high, it will automatically suppress video during the retrace periods when BLANK is high. The user may also pulse this input while BLANK is high to selectively suppress raster video.
AGREENF/ BKGND	17	I	**Green Foreground/Background Intensity:** In color mode, this input activates the GREEN/GP1 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input specifies gray or black screen background.
ABLUEF/ ABLANK	23	I	**Blue Foreground/Blank Attribute:** In color mode, this input activates the BLUE/TTLV2 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input generates a blank space for the associated character. The blank space intensity is controlled by the AGREENF/BKGND input, the reverse video attribute and cursor input.
AREDF/AHILT	25	I	**Red Foreground/Highlight Attribute:** In color mode, this input activates the RED/TTLV1output during the foreground (character video) portion of the associated character block. In monochrome mode, this input highlights the associated character (including underline).
CURSOR	8	I	**Cursor Timing:** This input provides the timing for the cursor video. In color mode, with CURSOR and CMODE high, the RGB outputs are driven high (white cursor). If CMODE is low, or in monochrome mode, this input reverses the intensities of the video and attributes (the foreground and background intensities are reversed). Cursor position, shape, and blink rate are controlled by this input.
CMODE	9	I	**Cursor Mode:** Used in color mode only. When CURSOR and CMODE are high, the RGB outputs are driven high (white cursor). When CURSOR is high and CMODE is low, the RGB outputs are logically inverted (reverse video cursor).
AUL	19	I	**Underline Attribute:** Specifies a line to be displayed in the character block. The specific line(s) are specified by the UL input. All other attributes apply to the underline video.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO. DIP	TYPE	NAME AND FUNCTION
ABLINK	16	I	**Blink Attribute:** In color mode, this active high input will drive the foreground RGB combination to the background RGB combination. In monochrome mode, the associated character or background is driven to the intensity determined by BKGND, reverse video attribute and the cursor input.
ADOUBLE	14	I	**Double Width Attribute:** This active high input causes the associated character video to be shifted out of the serial shift register at one half the dot frequency (DCLK). The $\overline{\text{CCLK}}$ output is not affected.
AREDB/AGP1	21	I	**Red Background/General Purpose Attribute 1:** In color mode, this input activates the RED/TTLV1 output during the background portion of the associated character block. In monochrome mode, it activates the GREEN/GP1 output for the associated character block.
ABLUEB/AGP2	22	I	**Blue Background/General Purpose Attribute 2:** In color mode, this input activates the BLUE/TTLV2 output during the background portion of the associated character block. In monochrome mode, it activates the LUM/GP2 output for the associated character block.
AGREENB/ ARVID	18	I	**Green Background/Reverse Video Attribute:** In color mode, this input activates the GREEN/GP1 output during the background portion of the associated character block. In monochrome mode, it causes the associated character block video intensities to be reversed.
D0 – D8	39, 2, 38, 3, 37, 4, 36, 5, 6	I	**Dot Data Input:** These are parallel inputs corresponding to the character/graphic symbol dot data for a given scan line. These inputs are strobed into the video shift register on the trailing (falling) edge of each character clock ($\overline{\text{CCLK}}$).
C0, C1	34, 35	I	**Character Clock Control:** The states of these two static inputs determine the internal divide factor for the $\overline{\text{CCLK}}$ output rate.
RESET	15	I	**Reset:** This active high input initializes the internal logic and resets the attribute latches.
M/C	30	I	**Monochrome/Color Mode:** This input selects whether the CMAC operates in monochrome or color mode. A low selects color mode and a high selects monochrome mode.
ADOTM	24	I	**Dot Modulation Attribute:** When DOTM and this input are high, the active dot width of the associated character video is one DCLK. When DOTM is high and this input is low, the active dot width of the associated character video is two DCLKs.
DOTM	31	I	**Dot Width Modulation:** When this input is high, two DCLKs are used for each dot shifted through the shift register. When this input is low, one DCLK is used.
DOTS	10	I	**Dot Stretching:** Sampled at the falling edge of BLANK. When this input is high, one extra dot is appended to individual dots or groups of dots of the input parallel data and then transferred through the shift register. When this input is low, normal transfer of input parallel data results.

FUNCTIONAL DESCRIPTION

The Turbo-CMAC consists of seven major sections (see block diagram). The high speed dot clock input is applied to a programmable divider to provide a character clock output for system timing. Parallel dot data is loaded into the video shift register on character boundaries and shifted into the video logic block at the dot rate specified by the dot modulation section. The appropriate attribute control inputs are selected by the mode select logic, latched internally on character boundaries, and combined with the serial dot data to provide monochrome or color video outputs.

The BLANK input defines the active screen and retrace areas. In color mode, the video outputs are forced to the specified background color when this signal is asserted; in monochrome mode the video outputs are forced to the states defined by the BKGND input; i.e., black if dark background is selected and gray if light background is selected. A separate RBLANK input allows the user to select the amount of border around the active area when operating in color mode or in monochrome mode with light background. This input can be tied high, in which case the area outside the active area will be dark, or it may be pulsed during BLANK periods to externally control the border widths.

In color mode, eight colors for the character (foreground) and eight colors for the background (area other than character) can be selected by the attribute inputs. In monochrome mode, the intensities of foreground and background are a function of the attribute and BKGND inputs; i.e., characters may be black, gray, white, or highlight (very white) while background may be black, gray, or white (see Table 1).

Character Clock Counter

The character clock counter divides the DCLK input to generate the character clock ($\overline{\text{CCLK}}$). The divide factor is specified by the clock control inputs (C1 – C0) as follows:

C1	C0	SCB2675T Dots/ Char.	SCB2675T $\overline{\text{CCLK}}$ Duty Cycle*
0	0	10	5/5
0	1	7	3/4
1	0	8	4/4
1	1	9	4/5

NOTE:
*High/low

The number of dot clocks/character is normally the number of dots/character as listed above. However, when dot width control is specified, the DCLK input is divided by two before it is applied to the character clock counter resulting in the number of dot clocks/character being double those listed above, although the number of displayed dots/character remains the same. (See Dot Modulation Logic section). The number of dots per character (C1 – C0) can be dynamically changed on either edge of the character clock as shown in Figure 10.

Table 1. Monochrome Mode Attribute Characteristics

REV[1]	AHILT	ABLINK[2]	FOREGROUND VIDEO	BACKGROUND VIDEO
0	0	0	W	B
0	0	1	W/G	B
0	1	0	H	B
0	1	1	H/W	B
1	0	0	B	G
1	0	1	B/W	G/B
1	1	0	B	W
1	1	1	B/H	W/B

NOTES:

1. REV = (BKGND) XOR (ARVID):

BKGND	ARVID	REV
0	0	0
0	1	1
1	0	1
1	1	0

2. For blinking, the video outputs are shown as 0/1, where 0 and 1 are the blink timing output states.
3. Foreground includes underline when underlining is specified by AUL = 1.
4. When ABLANK = 1, foreground component becomes same as background component.
5. Codes for video outputs are as follows:

CODE	TTLV2	TTLV1	BEAM INTENSITY
B	0	0	Black (B)
G	0	1	Gray (G)
W	1	0	White (W)
H	1	1	Highlight (H)

Video Shift Register

On each character boundary, the parallel input dot data (D0 – D8) is loaded into the video shift register. The data is shifted out least significant bit first (D0) at the DCLK rate. If 10 dots/character are specified (C1 – C0 = 00), the tenth dot will be the same as D8. The serial dot data from the video shift register is routed to the video logic where it is combined with the cursor and attribute control bits to produce the video data outputs.

Mode Select, Attribute and Cursor Control

The mode select logic multiplexes the monochrome and color attribute inputs and outputs as specified by the M/C input. The monochrome mode provides blank, reverse video, highlight and two general purpose attributes. The latter may be used, with external logic, to combine other attributes (e.g., overscore) into the video stream. The color mode provides RGB foreground and background color attributes. Both modes provide double width characters, blink, underline, dot width control and dot stretching.

The cursor and attribute inputs are pipelined internally to allow for system pipeline propagations. The cursor input and the attribute inputs are delayed for one $\overline{\text{CCLK}}$ to account for the delay of the character data through the character generator latches. The attribute timing inputs (BLINK, UL and DOTS) are clocked into the 2675 at the beginning of each scan line time by the falling edge of BLANK. Thus, these inputs must be in their proper state at the falling edge of BLANK preceding the scan line where they are required to be active. The BLANK signal itself is also delayed internally to provide for the RAM and character generator delays (see Figures 6 and 7). Internal delays cause the video outputs to be delayed relative to $\overline{\text{CCLK}}$ as illustrated in Figure 8.

Video Logic

Each character block consists of the three components shown in Figure 1. Symbol video is generated from the dot data inputs D0 – D8. Underline video is enabled by the AUL attribute and is generated during the scan lines for which the UL input is active. Underline and symbol video are always the same intensity or color, and other attributes (e.g., ABLINK) apply to them equally. The combination of underline and symbol video is also referred to as foreground video. Background video is the area of the character block corresponding to the absence of foreground video. The assertion of the nondisplay attribute (ABLANK) in the monochrome mode causes the entire character block to be displayed as background.

In monochrome mode, the serial dot data and pipelined cursor and attributes are combined to generate four video intensities (black, gray, white and highlight) which are encoded on the TTLV1 and TTLV2 outputs as follows:

TTLV2	TTLV1	VIDEO INTENSITY
0	0	Black
0	1	Gray
1	0	White
1	1	Highlight

Table 1 describes the relationship between attributes and video intensity of the foreground and background components of the character block in monochrome mode.

In color mode, the colors of the foreground and background components are specified by the corresponding attribute inputs; AREDF, AGREENF and ABLUEF dictate the color of the foreground component while AREDB, AGREENB and ABLUEB do the same for the background component. In this mode, the serial dot data and pipelined cursor and attributes are combined to generate four video outputs. The RED, GREEN and BLUE outputs separately contain the corresponding foreground and background components. The LUM output is the logical-OR of the foreground colors and can be used to drive a separate monochrome monitor or to select a different set of colors for the foreground.

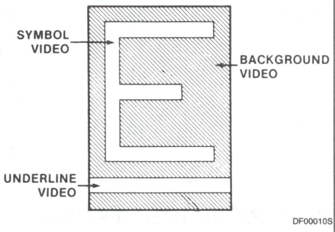


Figure 1. Character Block Definition

Dot Modulation Logic

The dot modulation logic controls the video shift register to supply dot stretching and dot width control.

Dot stretching is controlled by the DOTS input which is sampled each scan line at the trailing (falling) edge of BLANK. If DOTS is asserted at that time, all characters on the following scan line will have dot stretching applied. Dot stretching causes an extra dot to be added to individual dots or groups of dots as shown in Figures 2 and 3. Dot stretching can be used to:

1. Compensate for low video bandwidth monitors (since the minimum active displayed segment with dot stretching is two DCLKs).
2. Assure crisp black characters when operating in white background mode.
3. Provide thick characters as a means of distinguishing areas of the display.

Turbo Color/Monochrome Attributes Controller (Turbo-CMAC) SCB2675T

Dot width is controlled by the DOTM and ADOTM inputs. DOTM is tied either high, which enables the feature on the entire display, or low, which disables the feature. With ADOTM high, the dot width of characters can be selectively controlled by assertion of the ADOTM attribute input. When operating in this mode, the dot clock input is divided by two before being applied to other circuits in the Turbo-CMAC.

The $\overline{\text{CCLK}}$ output is also divided by two. The truth table for the possible combinations of ADOTM and DOTM is as follows:

ADOTM	DOTM	OPERATION
0	0	Normal mode
0	1	Dot width control, 100% duty cycle (DCLK÷2, $\overline{\text{CCLK}}$÷2)
1	0	Not allowed
1	1	Dot width control, 50% duty cycle (DCLK÷2, $\overline{\text{CCLK}}$÷2)

When dot width control is enabled as above, two DCLKs are used for each video dot period. Asserting ADOTM for a particular character will cause each active video dot of the displayed character to be turned on for one DCLK and off for the other DCLK, while if ADOTM is negated for that character, the active video dot for that character will be turned on (black background) or off (white background) for both DCLK times (see figures 2 and 4). Only the character video component of the character block is modulated. Underline video and background are not affected by on-time modulation. Width control can be used to:

1. Make horizontal lines and vertical lines appear the same brightness on the display.
2. Provide two different brightness levels for characters without requiring a monitor with analog brightness inputs.

However, note that the effects produced by this feature are highly dependent on the video amplifier characteristics of the monitor used.

Double Width Logic

The double width logic controls the rate at which dots are shifted through the video shift register. When the ADOUBLE input is asserted, the associated character video will be shifted at one half the DCLK rate, and the dot information for the next character will be loaded into the shift register two $\overline{\text{CCLK}}$s later. The character and attribute data that is present on the first $\overline{\text{CCLK}}$ after the ADOUBLE input is asserted will be ignored. The $\overline{\text{CCLK}}$ output is not affected. If a double width character is specified at the last location of a character row, the second half of the double width character (one $\overline{\text{CCLK}}$) will extend into the horizontal front porch.

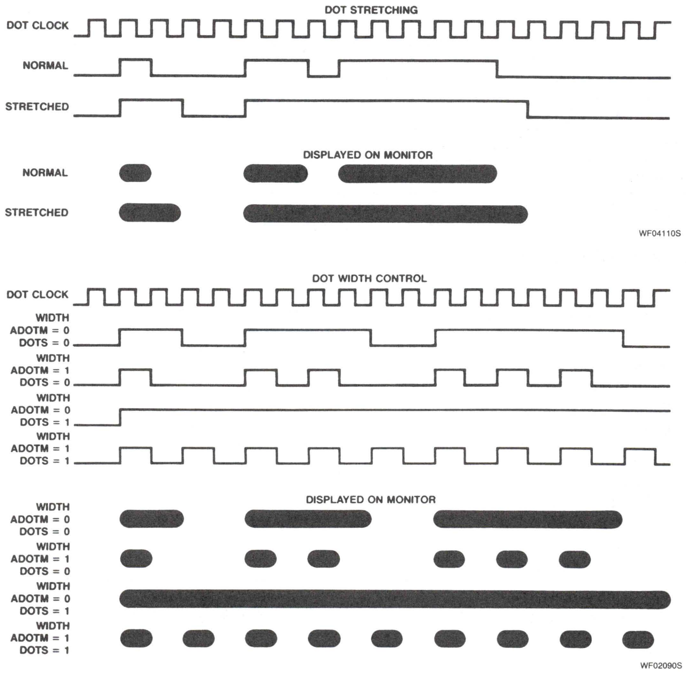

DOT STRETCHING
DOT CLOCK
NORMAL
STRETCHED
DISPLAYED ON MONITOR
NORMAL
STRETCHED
WF04110S
DOT WIDTH CONTROL
DOT CLOCK
WIDTH ADOTM = 0 DOTS = 0
WIDTH ADOTM = 1 DOTS = 0
WIDTH ADOTM = 0 DOTS = 1
WIDTH ADOTM = 1 DOTS = 1
DISPLAYED ON MONITOR
WIDTH ADOTM = 0 DOTS = 0
WIDTH ADOTM = 1 DOTS = 0
WIDTH ADOTM = 0 DOTS = 1
WIDTH ADOTM = 1 DOTS = 1
WF02090S

Figure 2. Dot Modulation Timing

Turbo Color/Monochrome Attributes Controller (Turbo-CMAC) SCB2675T

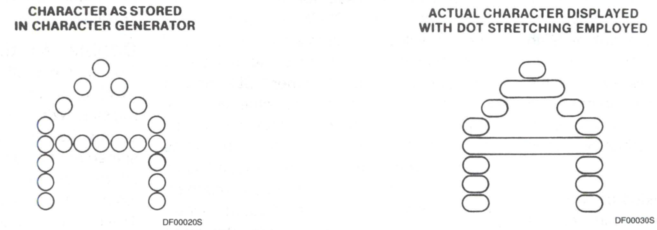

CHARACTER AS STORED
IN CHARACTER GENERATOR
ACTUAL CHARACTER DISPLAYED
WITH DOT STRETCHING EMPLOYED
DF00020S
DF00030S

Figure 3. Dot Stretching

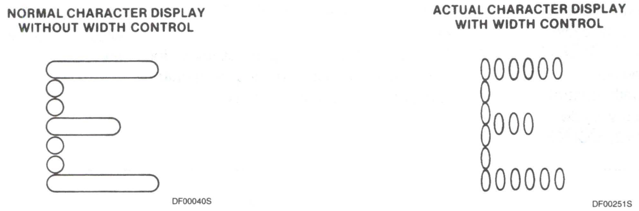

NORMAL CHARACTER DISPLAY
WITHOUT WIDTH CONTROL
ACTUAL CHARACTER DISPLAY
WITH WIDTH CONTROL
DF00040S
DF00251S

Figure 4. Dot Width Control

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{SRG}	Storage temperature range	-65 to +150	°C
	All voltages with respect to ground[3]	-0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$, V_{BB} = Figure 5[4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{IL}	Input low voltage			0.8	V
V_{IH}	Input high voltage		2		V
V_{OL}	Output low voltage	$I_{OL} = 4mA$		0.4	V
V_{OH}	Output high voltage	$I_{OH} = -400\mu A$	2.4		V
I_{IL}	Input low current	$V_{IN} = 0.4V$			
	DCLK			-800	µA
	All other inputs			-400	µA
I_{IH}	Input high current	$V_{IN} = 2.4V$			
	DCLK			40	µA
	All other inputs			20	µA
I_{CC}	V_{CC} supply current	$V_{IN} = 0V$, V_{CC} = Max		30	mA
I_{BB}	V_{BB} supply current	Figure 5		110	mA

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$, V_{BB} = Figure 5[4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	TENTATIVE LIMITS 40MHz Version Min	TENTATIVE LIMITS 40MHz Version Max	UNIT
Dot clock timing (Figure 8)					
f_D	Frequency			40	MHz
t_{DH}	High time		10		ns
t_{DL}	Low time		10		ns
Setup times (Figures 6, 7, 9 and 10)					
t_{SB}	BLANK to $\overline{\text{CCLK}}$		35		ns
t_{SA}	Attributes to $\overline{\text{CCLK}}$		40		ns
t_{SD}	D0–D9 to $\overline{\text{CCLK}}$		50		ns
t_{SK}	CURSOR to $\overline{\text{CCLK}}$		35		ns
t_{SC}	C0, C1 to DCLK		15		ns
t_{SR}	RBLANK to DCLK		15		ns
t_{SM}	BLINK, UL, DOTS to BLANK		10		ns
Hold times (Figures 6, 7, 9 and 10)					
t_{HB}	BLANK from $\overline{\text{CCLK}}$		5		ns
t_{HA}	Attributes from $\overline{\text{CCLK}}$		5		ns
t_{HD}	D0-D8 from $\overline{\text{CCLK}}$		5		ns
t_{HK}	CURSOR from $\overline{\text{CCLK}}$		5		ns
t_{HC}	C0, C1 from DCLK		5		ns
t_{HR}	RBLANK from DCLK		10		ns
t_{HM}	BLINK, UL, DOTS from BLANK		15		ns
Delay times (Figure 8)					
t_{DC}	$\overline{\text{CCLK}}$ from DCLK	$C_L = 50pF$		45	ns
t_{DV}	Other outputs from DCLK			45	ns

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. For testing, all input signals swing between 0.4V and 2.4V with a transition time of 3ns maximum. All time measurements are referenced at input voltages of 0.8V and 2V and output voltages are checked at 0.8V and 2V.

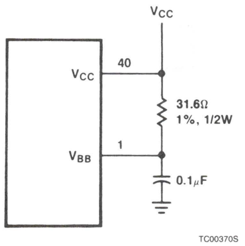
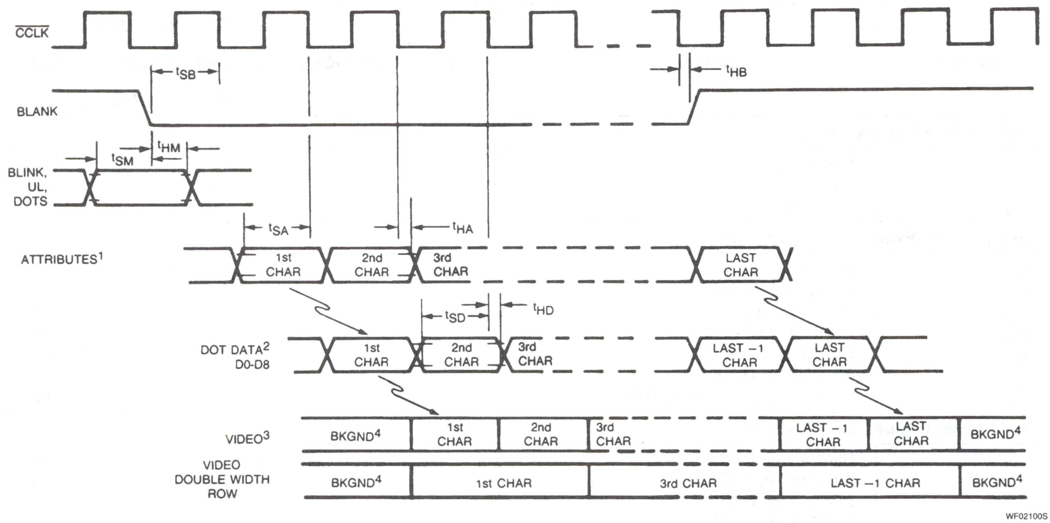


Figure 5. Recommended V_{BB} Test Circuit

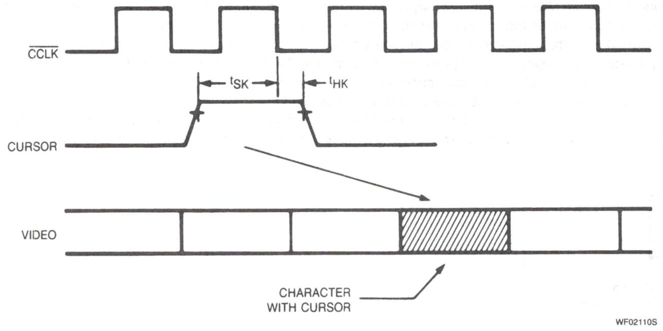


$\overline{\text{CCLK}}$
t_{SB}
t_{HB}
BLANK
t_{HM}
t_{SM}
BLINK, UL, DOTS
t_{SA}
t_{HA}
ATTRIBUTES¹
1st CHAR
2nd CHAR
3rd CHAR
LAST CHAR
t_{SD}
t_{HD}
DOT DATA² D0-D8
1st CHAR
2nd CHAR
3rd CHAR
LAST −1 CHAR
LAST CHAR
VIDEO³
BKGND⁴
1st CHAR
2nd CHAR
3rd CHAR
LAST − 1 CHAR
LAST CHAR
BKGND⁴
VIDEO DOUBLE WIDTH ROW
BKGND⁴
1st CHAR
3rd CHAR
LAST −1 CHAR
BKGND⁴
WF02100S

NOTES:

1. Attributes include ABLINK, ABLANK, ARVID, AUL, AHILT, ADOUBLE, ADOTM, two general purpose, and foreground/background colors.
2. One $\overline{\text{CCLK}}$ delay for dot data (obtained from delay through character generator).
3. See Figure 8 for detail timing of video outputs.
4. Non-active scan time. VIDEO reverts to polarity selected by the BKGND input in monochrome mode.

Figure 6. Turbo-CMAC Pipeline Timing



$\overline{\text{CCLK}}$
t_{SK}
t_{HK}
CURSOR
VIDEO
CHARACTER WITH CURSOR
WF02110S

Figure 7. Cursor Pipeline Timing

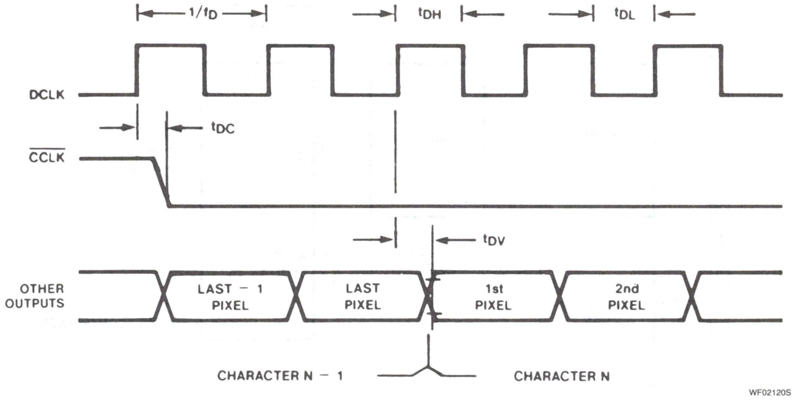

1/fD
tDH
tDL
DCLK
tDC
CCLK
tDV
OTHER OUTPUTS
LAST − 1 PIXEL
LAST PIXEL
1st PIXEL
2nd PIXEL
CHARACTER N − 1
CHARACTER N
WF02120S

Figure 8. Output Pipeline Timing

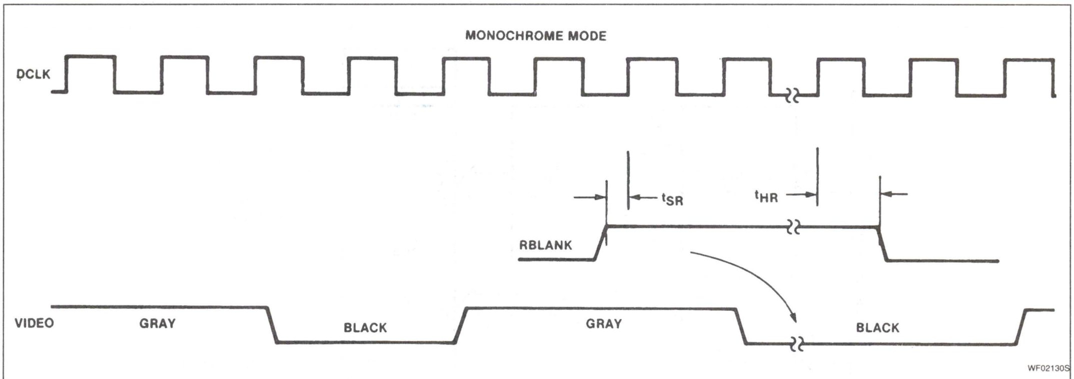
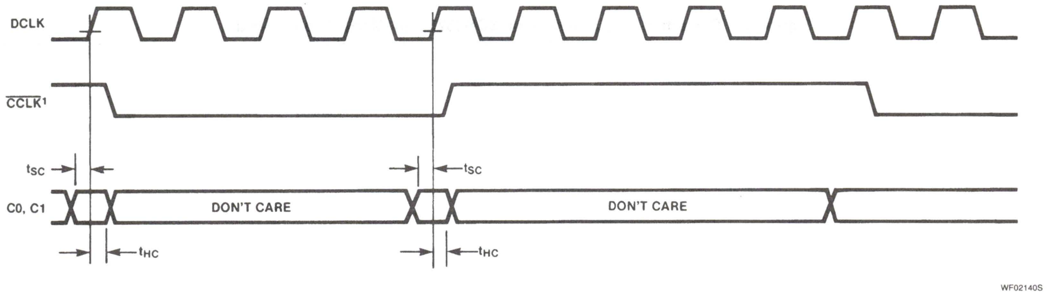

MONOCHROME MODE
DCLK
tSR
tHR
RBLANK
VIDEO
GRAY
BLACK
GRAY
BLACK
WF02130S

Figure 9. RBLANK Timing During Inactive Scan Time (Blank = 1)


DCLK
CCLK¹
tSC
tSC
C0, C1
DON'T CARE
DON'T CARE
tHC
tHC
WF02140S

NOTE:

1. The high and low times of $\overline{\text{CCLK}}$ may be controlled independently.

Figure 10. Clock Divider Timing

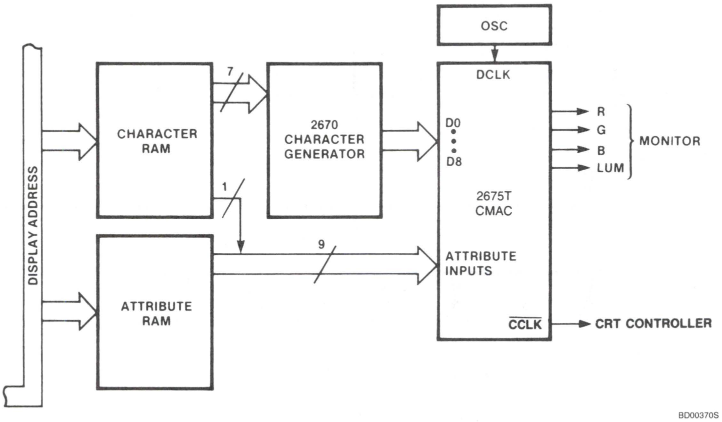

OSC
DCLK
CHARACTER RAM
7
2670 CHARACTER GENERATOR
D0
D8
R
G
B
LUM
MONITOR
DISPLAY ADDRESS
1
2675T CMAC
9
ATTRIBUTE INPUTS
ATTRIBUTE RAM
CCLK
CRT CONTROLLER
BD00370S

Figure 11. System Block Diagram of SCB2675T in Color Mode

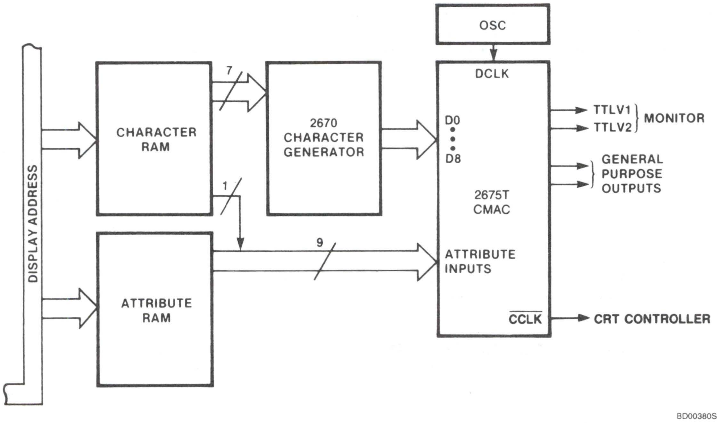

OSC
DCLK
CHARACTER RAM
7
2670 CHARACTER GENERATOR
D0
D8
TTLV1
TTLV2
MONITOR
GENERAL PURPOSE OUTPUTS
DISPLAY ADDRESS
1
2675T CMAC
9
ATTRIBUTE INPUTS
ATTRIBUTE RAM
CCLK
CRT CONTROLLER
BD00380S

Figure 12. System Block Diagram of SCB2675T in Monochrome Mode

Signetics

SCB2677
Video Attributes Controller (VAC)

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics SCB2677A and SCB2677B Video Attributes Controllers (VAC) are bipolar LSI devices designed for CRT terminals and display systems that employ raster scan techniques. Each contains a high-speed video shift register, field and character attributes logic, attribute latch, cursor format logic and half-dot shift control.

The VAC provides control of visual attributes on a field or character-by-character. Internal logic preserves field attribute data from character row to character row so that an attribute byte is not required at the beginning of each row. The SCB2677B provides for reverse video, blank (non-display), blink, underline and highlight attributes and a graphics mode attribute to work in conjunction with the Signetics SCN2670 Display Character and Graphics Generators (DCGG). The SCB2677A substitutes a strike-through attribute for the graphics attribute.

The horizontal dot frequency is the basic timing input to the VAC. Internally, this clock is divided down to provide a character clock output for system synchronization. Up to ten bits of video dot data are parallel-loaded into the video shift register on each character boundary. The video data is encoded to three levels of intensity (black, gray and white) and output on two TTL outputs. Light or dark screen background may be specified.

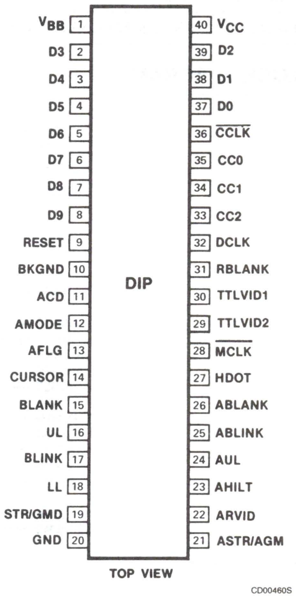
FEATURES

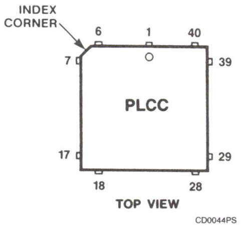
- **18MHz and 25MHz video dot rates**
- **Three level-encoded TTL video outputs**
- **Character/field attribute logic**
  - **Reverse video**
  - **Character blank**
  - **Character blink**
  - **Underline**
  - **Highlight**
- **Strike-through or graphics control**
- **Field attributes extend from row to row**
- **Light or dark field**
- **Cursor reverse video logic**
- **Up to 12 dots per character**
- **Retrace blanking for light field**
- **Optional field graphics control output**
- **High-speed bipolar design**
- **TTL compatible**
- **Compatible with Signetics SCN2672 PVTC, SCN2674 AVDC and SCN2670 DCGG**
- **Upgrade of the Signetics SCB2673 VAC**

APPLICATIONS

- **CRT terminals**
- **Word processing systems**
- **Small business computers**

PIN CONFIGURATIONS


VBB 1
D3 2
D4 3
D5 4
D6 5
D7 6
D8 7
D9 8
RESET 9
BKGND 10
ACD 11
AMODE 12
AFLG 13
CURSOR 14
BLANK 15
UL 16
BLINK 17
LL 18
STR/GMD 19
GND 20
40 VCC
39 D2
38 D1
37 D0
36 CCLK
35 CC0
34 CC1
33 CC2
32 DCLK
31 RBLANK
30 TTLVID1
29 TTLVID2
28 MCLK
27 HDOT
26 ABLANK
25 ABLINK
24 AUL
23 AHILT
22 ARVID
21 ASTR/AGM
DIP
TOP VIEW
CD00460S


INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

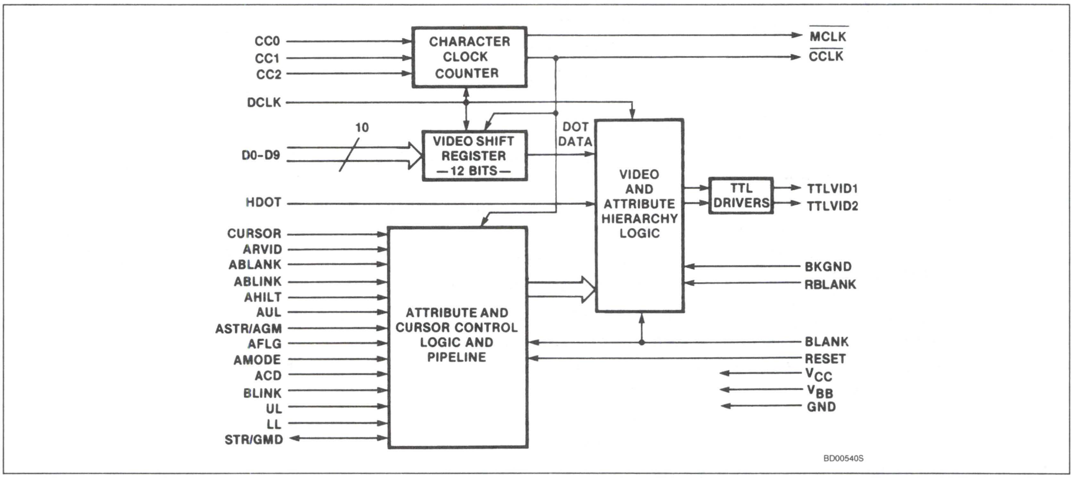
Pin	Function	Pin	Function	Pin	Function
1	NC	16	CURSOR	30	HDOT
2	V_{BB}	17	BLANK	31	$\overline{\text{MCLK}}$
3	D3	18	UL	32	TTLVID2
4	D4	19	BLINK	33	TTLVID1
5	D5	20	LL	34	NC
6	D6	21	STR/GMD	35	RBLANK
7	D7	22	GND	36	DCLK
8	D8	23	NC	37	CC2
9	D9	24	ASTR/AGM	38	CC1
10	RESET	25	ARVID	39	CC0
11	BKGND	26	AHILT	40	$\overline{\text{CCLK}}$
12	NC	27	AUL	41	D0
13	ACD	28	ABLINK	42	D1
14	AMODE	29	ABLANK	43	D2
15	AFLG			44	V_{CC}

Video Attributes Controller (VAC) SCB2677

ORDERING INFORMATION

PACKAGES	$V_{CC} = +5V \pm 5\%$, $T_A = 0°C$ to $+70°C$			
	Graphics Attribute		Strike-Through Attribute	
	25MHz	18MHz	25MHz	18MHz
Ceramic DIP	SCB2677BC5I40	SCB2677BC8I40	SCB2677AC5I40	SCB2677AC8I40
Plastic DIP	SCB2677BC5N40	SCB2677BC8N40	SCB2677AC5N40	SCB2677AC8N40
Plastic LCC	SCB2677BC5A44	SCB2677BC8A44	SCB2677AC5A44	SCB2677AC8A44

BLOCK DIAGRAM


CC0
CC1
CC2
CHARACTER CLOCK COUNTER
MCLK
CCLK
DCLK
10
D0-D9
VIDEO SHIFT REGISTER —12 BITS—
DOT DATA
VIDEO AND ATTRIBUTE HIERARCHY LOGIC
TTL DRIVERS
TTLVID1
TTLVID2
HDOT
CURSOR
ARVID
ABLANK
ABLINK
AHILT
AUL
ASTR/AGM
AFLG
AMODE
ACD
BLINK
UL
LL
STR/GMD
ATTRIBUTE AND CURSOR CONTROL LOGIC AND PIPELINE
BKGND
RBLANK
BLANK
RESET
V_{CC}
V_{BB}
GND
BD00540S

Video Attributes Controller (VAC)

SCB2677

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
DCLK	32	36	I	**Dot Clock:** Dot frequency input. Video output shift rate.
$\overline{\text{CCLK}}$	36	40	O	**Character Clock:** A submultiple of DCLK. The frequency ranges from one sixth to one twelfth of DCLK, as determined by the state of the CC0–CC2 inputs.
CC2 – CC0	33 – 35	37 – 39	I	**Character Clock Control:** The logic state on these three static inputs determine the internal divide factor for the $\overline{\text{CCLK}}$ output rate. Character clock rates of 6 through 12 dots per character may be specified.
D0 – D9	37 – 39, 2 – 8	41 – 43, 3 – 9	I	**Dot Data Input:** These are parallel inputs corresponding to the character/graphic symbol dot data for a given scan line. These inputs are strobed into the video shift register on the falling edge of each character clock.
HDOT	27	30	I	**Half-Dot Shift:** When this input is high, the serial video output is delayed by one-half dot time. This input is latched on the falling edge of each character clock.
CURSOR	14	16	I	**Cursor Timing:** This input provides the timing for the cursor video. When high, effectively reverses the intensities of the video and attributes. Cursor position, shape, and blink rate are controlled by this input.
BKGND	10	11	I	**Background Intensity:** Specifies light or dark video during BLANK and character fields. Affects the intensities of all attributes.
BLANK	15	17	I	**Screen Blank:** When high, this input forces the video outputs to the level specified by the BKGND input (either high or low intensity). Not effective when RBLANK is high.
RBLANK	31	35	I	**Retrace Blank:** This input is used to force the two video outputs to a low intensity (black) during retrace intervals. If held high (1), it will automatically suppress video when BLANK is high (1). The user may pulse this input while BLANK is high to selectively suppress raster video.
ARVID	22	25	I	**Reverse Video Attribute:** The intensity of the associated character or field video is reversed. All other attributes are effectively reversed.
AHILT	23	26	I	**Highlight Attribute:** All-dot video (including underline) of the associated character or field is highlighted with respect to the BKGND input and the reverse video attribute.
ABLANK	26	29	I	**Blank Attribute:** Generates a blank space in the associated character or field. The blank space intensity is determined by the BKGND input, the reverse video attribute, and the CURSOR input.
ABLINK	25	28	I	**Blink Attribute:** The associated character or field video is driven to the intensity determined by BKGND and the reverse video attribute when the BLINK input is high.
AUL	24	27	I	**Underline Attribute:** Specifies a line to be displayed on the character or field. The line is specified by the UL input. All other attributes apply to the underline video.
ASTR/AGM	21	24	I	**Strike-Through Attribute (2677A):** Specifies a line to be displayed on the character or field. The line is specified by the STR input. **Attribute Graphics Mode (2677B):** This input is latched and synchronized to provide a field GMD output for the SCN2670 DCGG.
AMODE	12	14	I	**Attribute Mode:** Specifies character (AMODE = 0) or field (AMODE = 1) attribute mode.
AFLG	13	15	I	**Attributes Flag:** The VAC samples and latches the attributes inputs when this input is high. If field attributes are specified (AMODE = 1), the attributes are double buffered on a row basis. Thus, each scan line of every character row will start with the attributes that were valid at the end of the previous row.
ACD	11	13	I	**Attribute Control Display:** In field attributes mode (AMODE = 1), if ACD = 0, the first character in each new attribute field (the attribute control character) will be suppressed and only the attributes will be displayed. If ACD = 1, the first character and the attributes are displayed. This input has no effect in character mode (AMODE = 0).
BLINK	17	19	I	**Blink:** This input is sampled on the falling edge of the BLANK to provide the blink rate for the character blink attribute. It should be a submultiple of the frame rate.
UL	16	18	I	**Underline:** Indicates the scan line(s) for the underline attribute. Latched on the falling edge of BLANK.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
STR/GMD	19	21	I O	**Strike-Through Line (2677A):** Indicates the scan line(s) for the strike-through attribute. Latched on the falling edge of BLANK. **Graphics Mode (2677B):** This output provides a synchronized, latched, field graphics mode corresponding to the AGM input. This output can be used to control the GM input on the SCN2670 DCGG.
LL	18	20	I	**Last Line:** Indicates the last scan line of each character row. Used internally to extend field attributes across row boundaries. Latched on the falling edge of BLANK. This input has no effect in character mode (AMODE = 0).
$\overline{MCLK}$	28	31	O	**Memory Clock:** This output is active for the last dot time for each $\overline{CCLK}$ period. See Figure 1.
TTLVID1	30	33	O	**TTL Video 1:** This output corresponds to the serial, non-highlighted video dot pattern.
TTLVID2	29	32	O	**TTL Video 2:** This output corresponds to the highlighted serial video dot pattern. Should be used with TTLVID1 to decode a composite video of three intensities.
RESET	9	10	I	**Manual Reset:** This active high input initializes the internal logic and resets the attribute latches.
V_{CC}	40	44	I	**Power Supply:** +5V.
V_{BB}	1	2	I	**Bias Supply:** See Figure 14.
GND	20	22	I	**Ground:** 0V reference.

FUNCTIONAL DESCRIPTION

The VAC consists of four major sections (see block diagram). The high-speed dot clock input is divided internally to provide a character clock for system timing. The parallel dot data is loaded into the video shift register on each character boundary and shifted into the video logic block at the dot rate. The six attribute inputs are latched internally and combined with the serial dot data to provide a three level video source for the monitor.

A separate BLANK input defines the active screen area. When BLANK = 0, the video levels are derived internally by the combinations of dot data, attributes, cursor, and the state of the BKGND input. Either black, gray or white background can be selected. Symbols (dot data) are normally gray and can be highlighted to white or black as shown in Figure 2.

During the inactive screen area (BLANK = 1), the video level produced by the TTL outputs is either gray (BKGND = 1) or black (BKGND = 0). A separate retrace blank (RBLANK) input is provided to suppress raster retrace video when gray background is specified. This input will force the video outputs to a low if RBLANK and BLANK = 1. The user may pulse RBLANK during the retrace interval in order to extend the gray border closer to the monitor edges.

Character Clock Counter

The character clock counter divides the frequency on the DCLK input to generate the character clock ($\overline{CCLK}$). The divide factor is specified by the clock control inputs (CC0 – CC2) as shown in the table above. See Figure 1.

CC2	CC1	CC0	$\overline{CCLK}$	
			Dots/Character	Duty Cycle*
0	0	0	6	3/3
0	0	1	6	3/3
0	1	0	7	4/3
0	1	1	8	4/4
1	0	0	9	5/4
1	0	1	10	5/5
1	1	0	11	6/5
1	1	1	12	6/6

NOTE:
*Low/high

Video Shift Register

On each character boundary, the parallel data (D0 – D9) is loaded into the video shift register. The data is shifted out least significant bit first (D0) by the DCLK. If 11 or 12 dots/character are specified (CC2 – CC0 = 110 or 111), a 0 (blank dot) is always shifted out before D0. For 12 dots/character, a 0 is also shifted out after D9. The serial dot data is shifted into the video logic where it is combined with the cursor and attributes to encode three levels of video.

Attribute And Cursor Control

The VAC visual attributes capabilities include: reverse video, character blank, blink, underline, highlight, and strike-through. The six attributes and the three attribute control inputs (AMODE, AFLG, and ACD) are clocked into the VAC on the falling edge of $\overline{CCLK}$. If AFLG is high, the attributes are latched internally and are effective for either one character time (AMODE = 0) or until another set of attributes is latched (AMODE = 1). The attributes set is double-buffered on a row-by-row basis internally. Using this technique, field attributes can extend across character row boundaries thereby eliminating the necessity of starting each row with an attribute set.

When field attribute mode is selected, (AMODE = 1), the VAC will accommodate two attribute storage configurations. In one configuration, the attribute control data is stored in the refresh RAM, taking the place of the first character code in the field to be affected. For this mode, the ACD input is tied low and blank characters will be displayed in the screen positions occupied by the attribute data (see Figure 12). The display RAM contains intermixed character and attribute data. When new attribute data is written to the SCB2677, the AFLG input is set high. The character at that location will be blanked, and only the attribute information will be displayed. That particular attribute data will be used for the resultant characters until the next AFLG pulse occurs. In the second configuration (ACD = 1), the character codes and attribute data are presented to the VAC in parallel (i.e., there are separate RAMs for the character and attribute data). In this mode, dot data is displayed at each character position (see Figure 13).

The CURSOR and the attribute input signals are pipelined internally to allow for system propagations (one $\overline{\text{CCLK}}$ for refresh RAM, one $\overline{\text{CCLK}}$ for dot generator). The attribute timing signals BLINK, UL, STR and LL are clocked into the VAC at the beginning of each scan line by the falling edge of the BLANK input. Thus, these signals must be in their proper state at the falling edge of BLANK preceding the scan line at which they are to be active (see Figure 5). The SCN2670 DCGG delays the character dot data by one character clock. The VAC assumes that there is a DCGG in the system and latches the dot data one character clock later than the latching of the attribute data.

Video Logic

The serial dot data and the pipelined cursor and attributes are combined to generate three levels (white, gray, and black) on two TTL compatible outputs, TTLVID1 and TTLVID2. The three levels are encoded as shown to the right.

The video is normally shifted out on the leading edge of the DCLK. When the HDOT input is asserted, the corresponding dot data is delayed by one-half DCLK. This half dot shifting, when used on selected lines of character video, can be used to effect eye-pleasing character rounding as shown in Figure 3. The half-dot shift does not extend into the next character's field boundary.

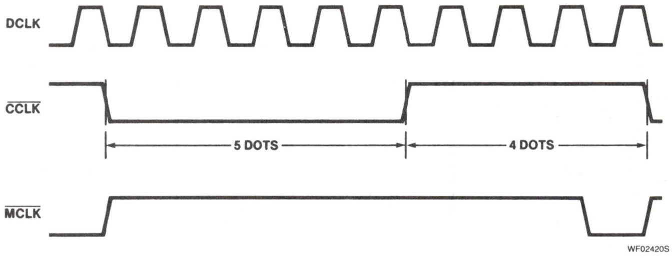


Figure 1. Clock Waveform (9 Dots/Character)

TTLVID2	TTLVID1	INTENSITY
0	0	Black
0	1	Gray
1	0	Not used
1	1	White

NOTE:
The TTLVID1 output can be used independently to generate a two-level non-highlighted video.

Attribute Hierarchy

The video of each character block consists of four components as shown in Figure 4.

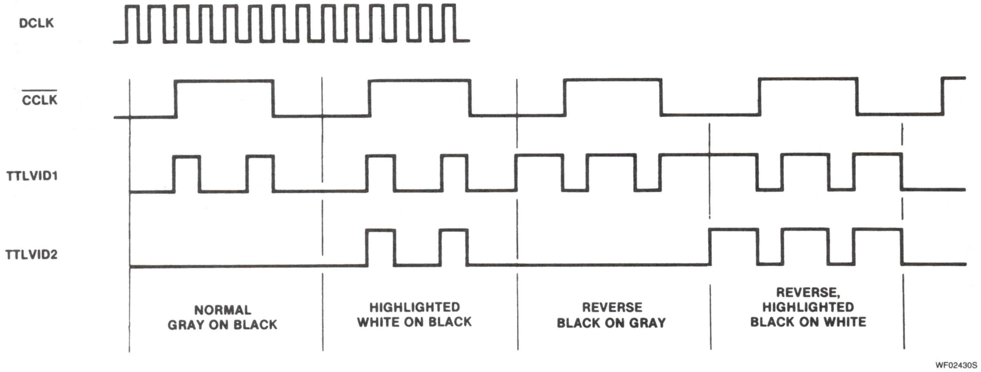
Symbol video is generated from the dot data inputs D0 – D9.

Underline video is enabled by the AUL attribute and is generated when the UL timing input is active. Underline and symbol video are always the same intensity.

Strike-through video is enabled by the ASTR attribute and is generated when the STR timing input is active. This video is the same intensity as the symbol and underline video. This feature applies to the SCB2677A only.

Surround video is the absence of symbol, underline and strike-through video or the presence of the non-display attributes (ABLANK or ABLINK • BLINK).

The relative intensities of the four video components are determined by the remaining attributes (AHILT, ABLANK, ABLINK, ARVID) and the BKGND and CURSOR inputs as illustrated in Table 1.



NOTES:
W = White
G = Gray
B = Black

Figure 2. Encoded Video Outputs

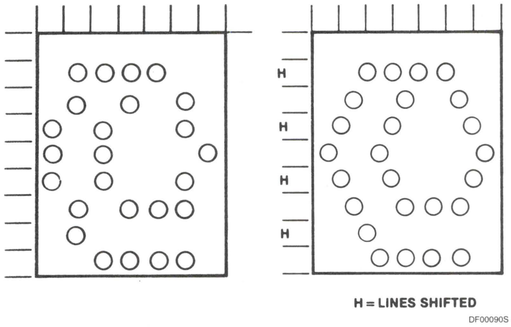

H
H
H
H
H = LINES SHIFTED
DF00090S

Figure 3. "AT" Symbol With and Without Half-Dot Shifting

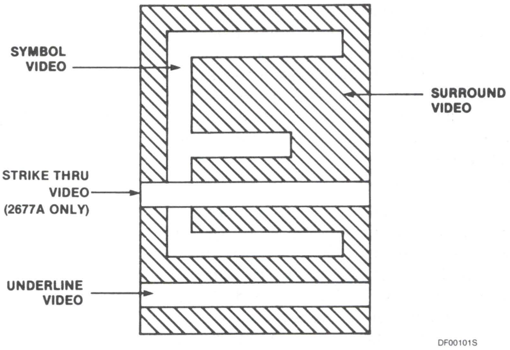

SYMBOL VIDEO
SURROUND VIDEO
STRIKE THRU VIDEO (2677A ONLY)
UNDERLINE VIDEO
DF00101S

Figure 4. Video Components of Character Block

Table 1. Attributes Hierarchy

BLANK	RBLANK	BKGND	REVERSE[1]	AHILT	"NON-" DISPLAY[2]	SYMBOL, UNDERLINE OR STRIKE-THROUGH[3, 4]	SURROUND VIDEO[3]
0	d	0	0	0	0	G	B
0	d	0	0	0	1	B	B
0	d	0	0	1	0	W	B
0	d	0	0	1	1	B	B
0	d	0	1	0	0	B	G
0	d	0	1	0	1	G	G
0	d	0	1	1	0	B	W
0	d	0	1	1	1	W	W
0	d	1	0	0	0	B	G
0	d	1	0	0	1	G	G
0	d	1	0	1	0	B	W
0	d	1	0	1	1	W	W
0	d	1	1	0	0	G	B
0	d	1	1	0	1	B	B
0	d	1	1	1	0	W	B
0	d	1	1	1	1	B	B
1	0	0	d	d	d	B	B
1	0	1	d	d	d	G	G
1	1	d	d	d	d	B	B

NOTES:
B = Black
G = Gray
W = White
d = Don't care
1. REVERSE = ARVID ⊕ CURSOR
2. Non-display = (ABLINK • BLINK) + ABLANK
3. See Figure 4.
4. Symbol, underline and strike-through are always same intensity.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	-65 to +150	°C
	All voltages with respect to ground	-0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$, V_{BB} = See Figure 14[3, 4, 5, 6]

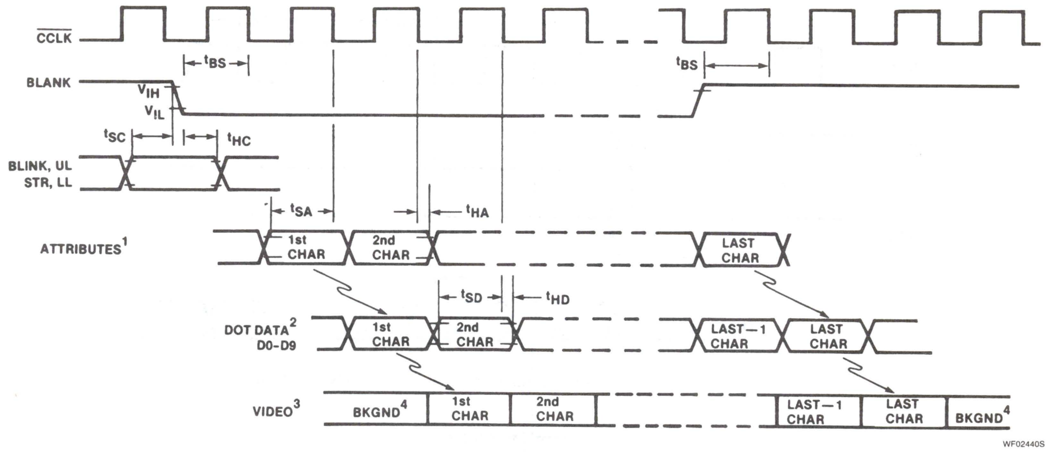
SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2			V
V_{OL}	Output low voltage	$I_{OL} = 4mA$			0.4	V
V_{OH}	Output high voltage	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input low current	$V_{IN} = 0.4V$			-400/ -800[6]	µA
I_{IH}	Input high current	$V_{IN} = 2.4V$			20/40[6]	µA
I_{CC}	V_{CC} supply current	$V_{IN} = 0V$, V_{CC} = Max			80	mA
I_{BB}	V_{BB} supply current	V_{BB} = Max			120	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. Operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. Parameters are valid over operating temperature range unless otherwise specified.
4. All voltage measurements are referenced to ground (V_{SS}). All input signals swing between 0.4V and 2.4V. All time measurements are referenced at input voltages of 0.8V, 2V and at output voltage of 0.8V, 2V as appropriate.
5. Typical values are at +25°C, typical supply voltages and typical processing parameters.
6. For DCLK input.
7. C_L less than 150pF minimum could be faster.

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$, V_{BB} = See Figure 14[3, 4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS				UNIT
			25MHz Version		18MHz Version		
			Min	Max	Min	Max	
Dot clock (Figure 11)							
f_D	Frequency			25		18	MHz
t_{DH}	High		15		22		ns
t_{DL}	Low		15		22		ns
Setup times to $\overline{\text{CCLK}}$ (Figures 5, 6, 7 and 11)							
t_{BS}	BLANK		50		50		ns
t_{SC}	BLINK, UL, STR, LL (ref to BLANK)		20		20		ns
t_{SA}	Attributes		45		55		ns
t_{SD}	Dot data D0 – D9		70		70		ns
t_{SK}	CURSOR		50		50		ns
t_{FS}	AFLG		50		65		ns
t_{SH}	HDOT		45		55		ns
Hold times from $\overline{\text{CCLK}}$ (Figures 5, 6, 7 and 11)							
t_{HC}	BLINK, UL, STR, LL (ref to BLANK)		20		20		ns
t_{HA}	Attributes		20		20		ns
t_{HD}	Dot data D0 – D9		30		30		ns
t_{HK}	CURSOR		20		20		ns
t_{FH}	AFLG		30		30		ns
t_{HH}	HDOT		20		20		ns
Setup times to DCLK (Figures 9,10)							
t_{SG}	BKGND		15		15		ns
t_{SB}	RBLANK		15		15		ns
t_{CS}	CC0 – CC2		30		35		ns
Hold times from DCLK (Figures 9,10)							
t_{HG}	BKGND		15		15		ns
t_{HB}	RBLANK		15		15		ns
t_{CH}	CC0 – CC2		20		20		ns
Delay times (Figures 7 and 8)							
t_{DGM}	GMD from DCLK			65		65	ns
t_{DC}	$\overline{\text{MCLK}}$, $\overline{\text{CCLK}}$ from DCLK	$C_L = 150pF$		65		65	ns
t_{DV}[7]	TTLVID1 and TTLVID2 from DCLK			75		80	ns


CCLK
tBS
BLANK
VIH
VIL
tSC
tHC
BLINK, UL
STR, LL
tSA
tHA
ATTRIBUTES1
1st CHAR
2nd CHAR
LAST CHAR
tSD
tHD
DOT DATA2
D0-D9
1st CHAR
2nd CHAR
LAST—1 CHAR
LAST CHAR
VIDEO3
BKGND4
1st CHAR
2nd CHAR
LAST—1 CHAR
LAST CHAR
BKGND4
WF02440S

NOTES:
1. Attributes include: ABLINK, ABLANK, ARVID, AUL, AHILT, and ASTR.
2. One $\overline{\text{CCLK}}$ delay for dot data (obtained from delay through character generator).
3. See Figure 8 for detail timing of TTLVID1,TTLVID2
4. Non-active scan time. VIDEO reverts to polarity selected by the BKGND input.

Figure 5. VAC Pipeline Timing

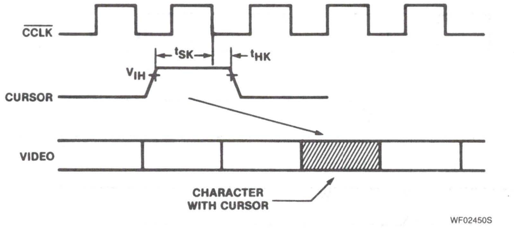
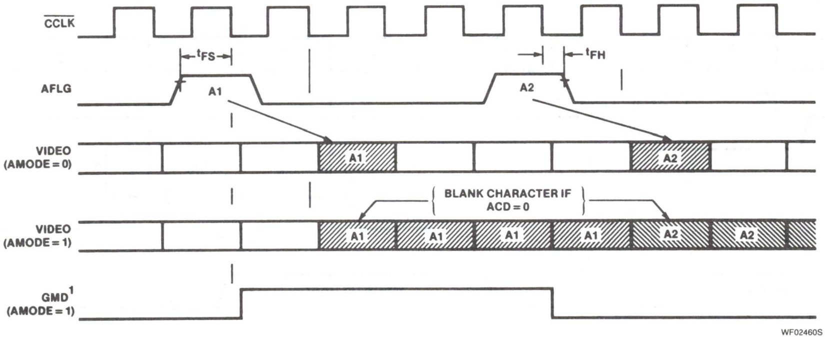
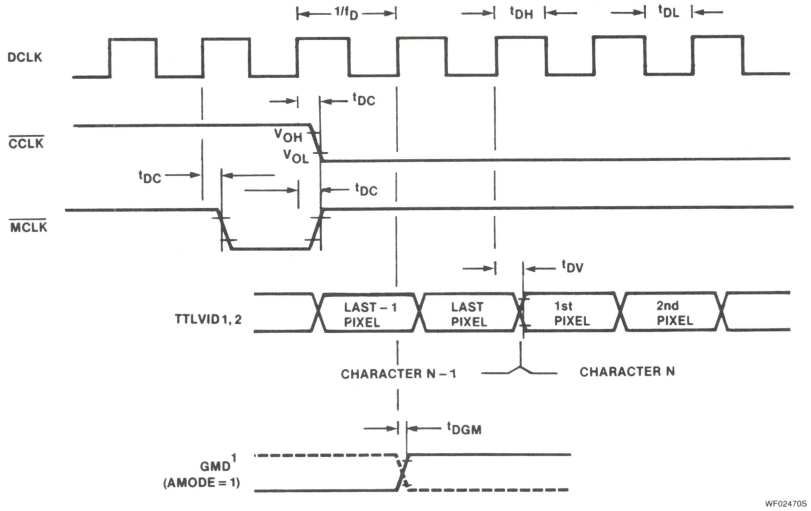

CCLK
tSK
tHK
VIH
CURSOR
VIDEO
CHARACTER WITH CURSOR
WF02450S

Figure 6. Cursor Pipeline Timing


CCLK
tFS
tFH
AFLG
A1
A2
VIDEO (AMODE = 0)
A1
A2
BLANK CHARACTER IF ACD = 0
VIDEO (AMODE = 1)
A1
A1
A1
A1
A2
A2
GMD1 (AMODE = 1)
WF02460S

NOTE:
1. GMD output in SCB2677B version only. See Figure 8 for detail timing.

Figure 7. Character (AMODE = 0), Field (AMODE = 1), and GMD Attribute Timing


$1/f_D$
t_{DH}
t_{DL}
DCLK
t_{DC}
$\overline{CCLK}$
V_{OH}
V_{OL}
t_{DC}
t_{DC}
$\overline{MCLK}$
t_{DV}
TTLVID 1, 2
LAST − 1 PIXEL
LAST PIXEL
1st PIXEL
2nd PIXEL
CHARACTER N − 1
CHARACTER N
t_{DGM}
GMD[1]
(AMODE = 1)
WF02470S

NOTE:
1. GMD output in SCB2677B version only.

Figure 8. Video and GMD Pipeline Timing

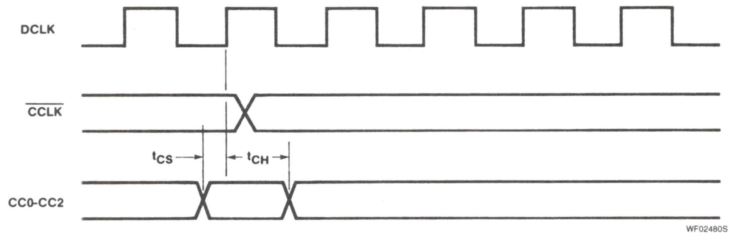

DCLK
$\overline{CCLK}$
t_{CS}
t_{CH}
CC0-CC2
WF02480S

Figure 9. Character Clock Counter Timing

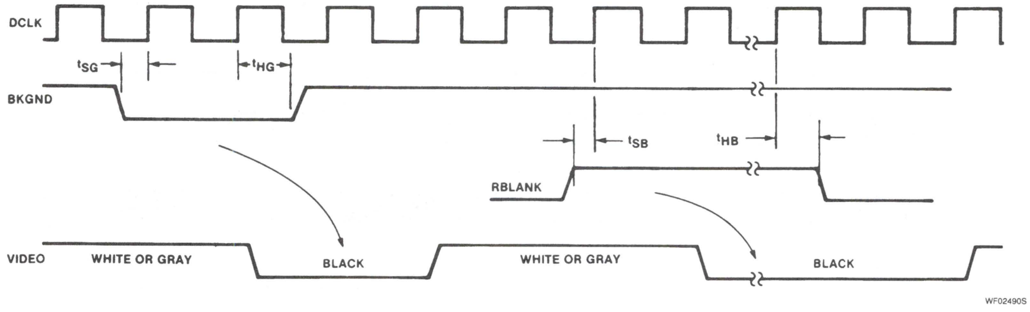

DCLK
t_{SG}
t_{HG}
BKGND
t_{SB}
t_{HB}
RBLANK
VIDEO
WHITE OR GRAY
BLACK
WHITE OR GRAY
BLACK
WF02490S

Figure 10. BKGND and RBLANK Timing During Inactive Scan Time (Blank = 1)

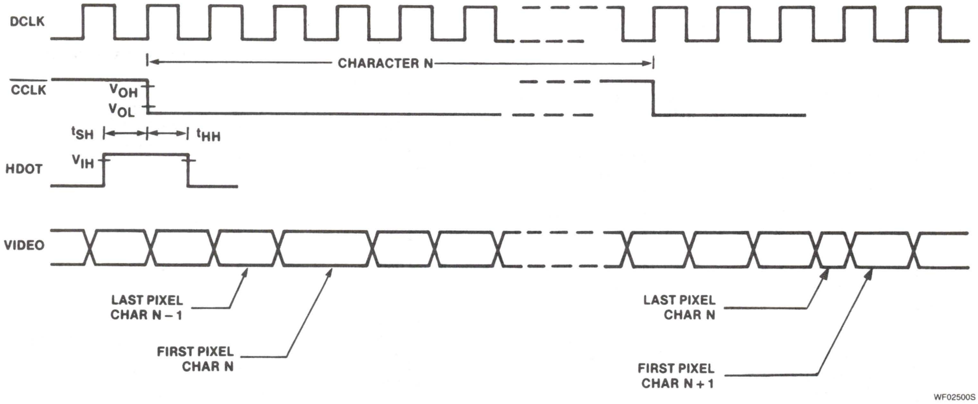

DCLK
CHARACTER N
CCLK
V_{OH}
V_{OL}
t_{SH}
t_{HH}
HDOT
V_{IH}
VIDEO
LAST PIXEL CHAR N − 1
FIRST PIXEL CHAR N
LAST PIXEL CHAR N
FIRST PIXEL CHAR N + 1
WF02500S

Figure 11. Half-Dot Shift Timing

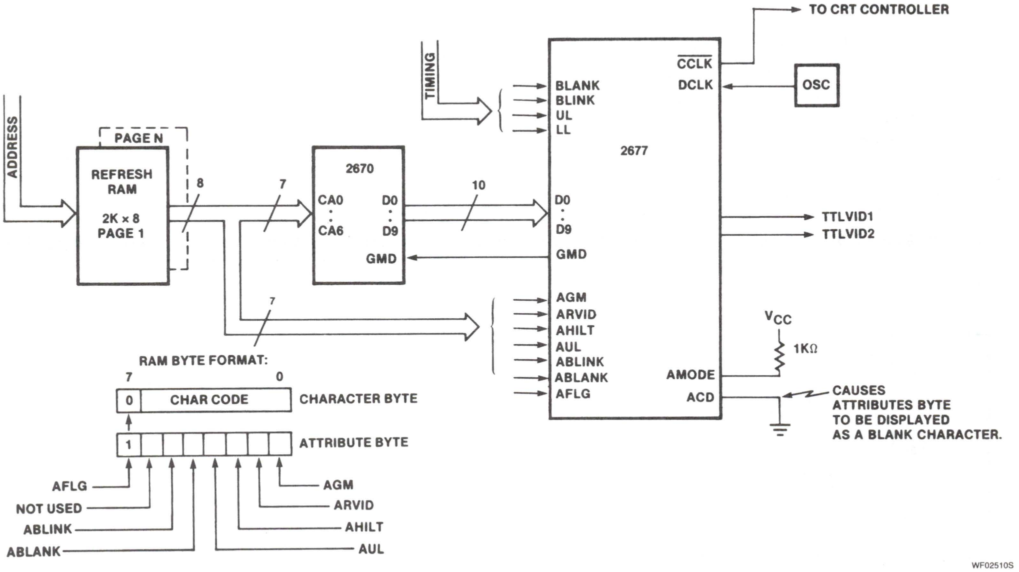
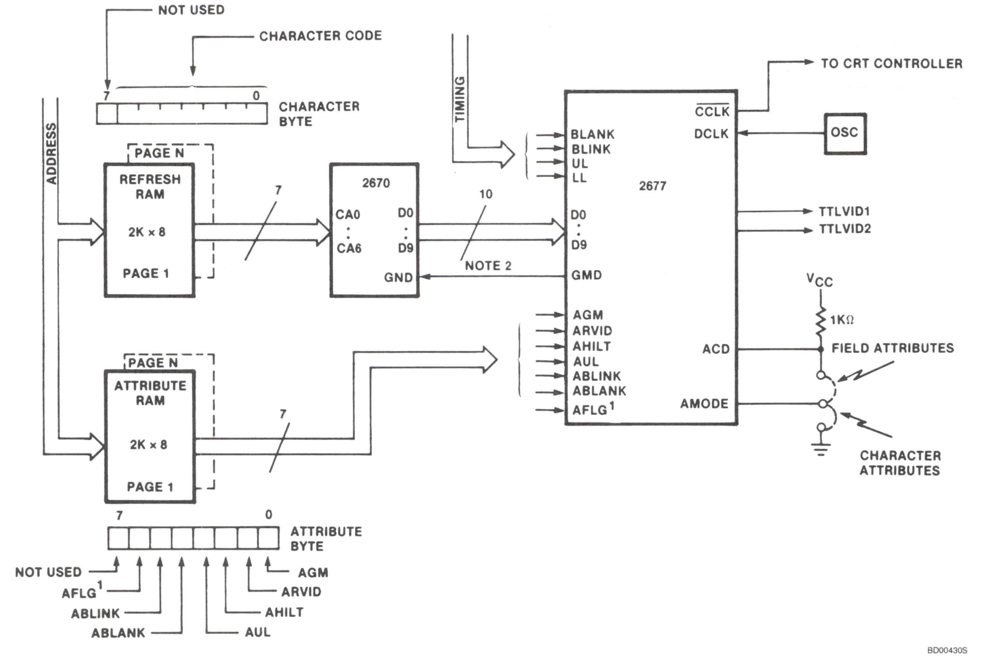

TO CRT CONTROLLER
TIMING
CCLK
DCLK
OSC
BLANK
BLINK
UL
LL
2677
ADDRESS
PAGE N
REFRESH RAM
2K × 8
PAGE 1
8
7
2670
CA0
CA6
D0
D9
10
D0
D9
GMD
GMD
TTLVID1
TTLVID2
7
AGM
ARVID
AHILT
AUL
ABLINK
ABLANK
AFLG
V_{CC}
1KΩ
AMODE
ACD
CAUSES ATTRIBUTES BYTE TO BE DISPLAYED AS A BLANK CHARACTER.
RAM BYTE FORMAT:
7
0
0
CHAR CODE
CHARACTER BYTE
1
ATTRIBUTE BYTE
AFLG
NOT USED
ABLINK
ABLANK
AGM
ARVID
AHILT
AUL
WF02510S

Figure 12. System Block Diagram of the 2677 in Field Attribute Mode Using the Narrow RAM (8-Wide) Configuration


NOT USED
CHARACTER CODE
7
0
CHARACTER BYTE
TIMING
ADDRESS
PAGE N
REFRESH RAM
2K × 8
PAGE 1
7
2670
CA0
CA6
D0
D9
GND
10
NOTE 2
TO CRT CONTROLLER
BLANK
BLINK
UL
LL
2677
CCLK
DCLK
OSC
D0
D9
GMD
TTLVID1
TTLVID2
VCC
1KΩ
FIELD ATTRIBUTES
AGM
ARVID
AHILT
AUL
ABLINK
ABLANK
AFLG1
ACD
AMODE
CHARACTER ATTRIBUTES
PAGE N
ATTRIBUTE RAM
2K × 8
PAGE 1
7
7
0
ATTRIBUTE BYTE
NOT USED
AFLG1
ABLINK
ABLANK
AGM
ARVID
AHILT
AUL
BD00430S

NOTE:
1. For operation in character attribute mode, tie AFLG high.
2. In character attribute mode, AGM output from RAM should be connected directly to the SCN2670 GMD input.

Figure 13. System Block Diagram of the 2677 in Field or Character Attribute Mode Using the Wide RAM Configuration

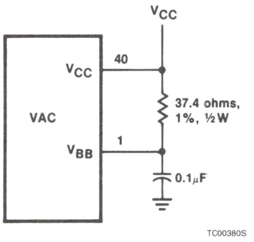

VCC
VAC
VCC
40
37.4 ohms, 1%, ½W
VBB
1
0.1μF
TC00380S

Figure 14. Test Diagram

Signetics

SCN2681

Dual Asynchronous Receiver/Transmitter (DUART)

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics SCN2681 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single-chip MOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It interfaces directly with microprocessors and may be used in a polled or interrupt driven system.

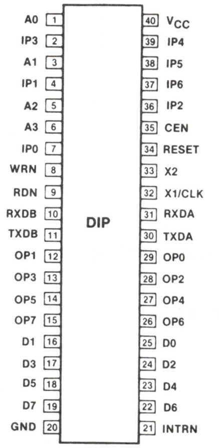
The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16× clock derived from a programmable counter/timer, or an external 1× or 16× clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

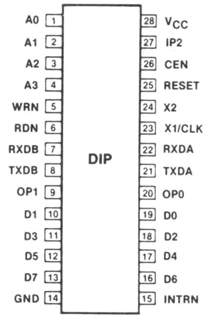
FEATURES

- **Dual full-duplex asynchronous receiver/transmitter**
- **Quadruple buffered receiver data registers**
- **Programmable data format**
  - **5 to 8 data bits plus parity**
  - **Odd, even, no parity or force parity**
  - **1, 1.5 or 2 stop bits programmable in 1/16-bit increments**
- **Programmable baud rate for each receiver and transmitter selectable from:**
  - **18 fixed rates: 50 to 38.4k baud**
  - **One user-defined rate derived from programmable timer/counter**
  - **External 1× or 16× clock**
- **Parity, framing, and overrun error detection**
- **False start bit detection**
- **Line break detection and generation**
- **Programmable channel mode**
  - **Normal (full-duplex)**
  - **Automatic echo**
  - **Local loopback**
  - **Remote loopback**
- **Multi-function programmable 16-bit counter/timer**
- **Multi-function 7-bit input port**
  - **Can serve as clock or control inputs**
  - **Change of state detection on four inputs**
- **Multi-function 8-bit output port**
  - **Individual bit set/reset capability**
  - **Outputs can be programmed to be status/interrupt signals**
- **Versatile interrupt system**
  - **Single interrupt output with eight maskable interrupting conditions**
  - **Output port can be configured to provide a total of up to six separate wire-ORable interrupt outputs**
- **Maximum data transfer: 1× – 1MB/s, 16× – 125kB/s**
- **Automatic wake-up mode for multidrop applications**
- **Start-end break interrupt/status**
- **Detects break which originates in the middle of a character**
- **On-chip crystal oscillator**
- **TTL compatible**
- **Single +5V power supply**

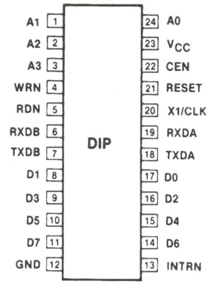
PIN CONFIGURATIONS


A0 1
IP3 2
A1 3
IP1 4
A2 5
A3 6
IP0 7
WRN 8
RDN 9
RXDB 10
TXDB 11
OP1 12
OP3 13
OP5 14
OP7 15
D1 16
D3 17
D5 18
D7 19
GND 20
DIP
40 V_{CC}
39 IP4
38 IP5
37 IP6
36 IP2
35 CEN
34 RESET
33 X2
32 X1/CLK
31 RXDA
30 TXDA
29 OP0
28 OP2
27 OP4
26 OP6
25 D0
24 D2
23 D4
22 D6
21 INTRN

CD00130S

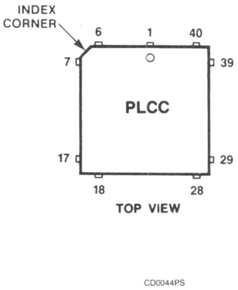

A0 1
A1 2
A2 3
A3 4
WRN 5
RDN 6
RXDB 7
TXDB 8
OP1 9
D1 10
D3 11
D5 12
D7 13
GND 14
DIP
28 V_{CC}
27 IP2
26 CEN
25 RESET
24 X2
23 X1/CLK
22 RXDA
21 TXDA
20 OP0
19 D0
18 D2
17 D4
16 D6
15 INTRN

CD00140S


A1 1
A2 2
A3 3
WRN 4
RDN 5
RXDB 6
TXDB 7
D1 8
D3 9
D5 10
D7 11
GND 12
DIP
24 A0
23 V_{CC}
22 CEN
21 RESET
20 X1/CLK
19 RXDA
18 TXDA
17 D0
16 D2
15 D4
14 D6
13 INTRN

CD00150S

PIN CONFIGURATIONS (Continued)



Pin	Function	Pin	Function
1	NC	23	NC
2	A0	24	INTRN
3	IP3	25	D6
4	A1	26	D4
5	IP1	27	D2
6	A2	28	D0
7	A3	29	OP6
8	IP0	30	OP4
9	WRN	31	OP2
10	RDN	32	OP0
11	RXDB	33	TXDA
12	NC	34	NC
13	TXDB	35	RXDA
14	OP1	36	X1/CLK
15	OP3	37	X2
16	OP5	38	RESET
17	OP7	39	CEN
18	D1	40	IP2
19	D3	41	IP6
20	D5	42	IP5
21	D7	43	IP4
22	GND	44	V_{CC}

Also provided on the SCN2681 are a multipurpose 7-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

The SCN2681 is available in four package versions: 40-pin and 28-pin, both 0.6″ wide DIPs; a compact 24-pin 0.4″ wide DIP; and a 44-pin PLCC.

ORDERING INFORMATION

PACKAGES	V_{CC} = +5V ±5%, T_A = 0°C to +70°C			
	24-Pin[1]	28-Pin[2]	40-Pin[2]	44-Pin
Ceramic DIP	Not available	SCN2681AC1I28	SCN2681AC1I40	Not available
Plastic DIP	SCN2681AC1N24	SCN2681AC1N28	SCN2681AC1N40	Not available
Plastic LCC	Not available	Not available	Not available	SCN2681AC1A44

NOTES:

1. 400 mil wide DIP
2. 600 mil wide DIP

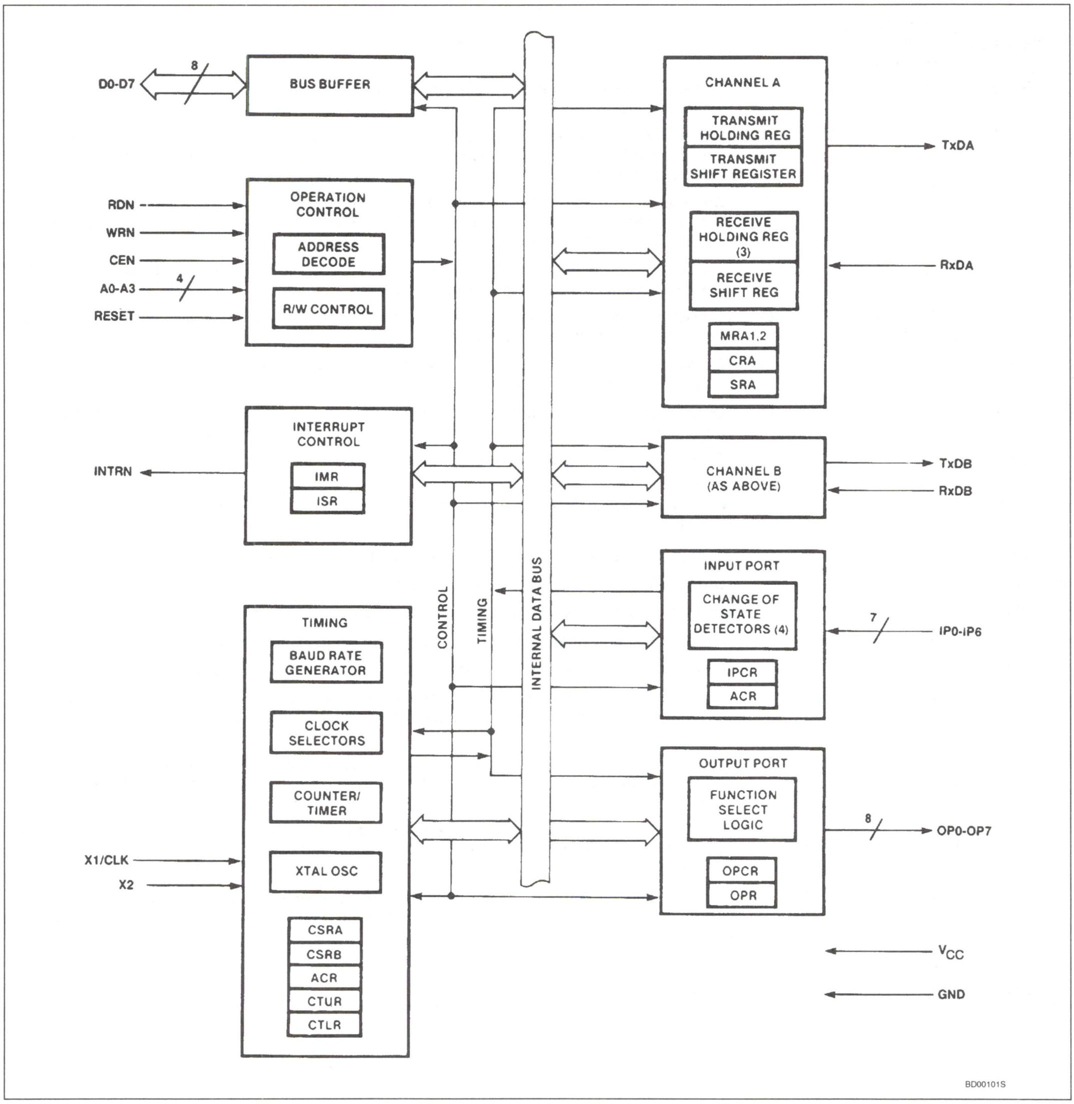
PIN DESCRIPTION

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
D0 – D7	X	X	X	I/O	**Data Bus:** Bidirectional 3-State data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CEN	X	X	X	I	**Chip Enable:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0 – D7 as controlled by the WRN, RDN and A0 – A3 inputs. When high, places the D0 – D7 lines in the 3-State condition.
WRN	X	X	X	I	**Write Strobe:** When low and CEN is also low, the contents of the data bus is loaded into the addressed register. The transfer occurs on the rising edge of the signal.
RDN	X	X	X	I	**Read Strobe:** When low and CEN is also low, causes the contents of the addressed register to be presented on the data bus. The read cycle begins on the falling edge of RDN.
A0 – A3	X	X	X	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESET	X	X	X	I	**Reset:** A high level clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), puts OP0 – OP7 in the high state, stops the counter/timer, and puts channels A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
INTRN	X	X	X	O	**Interrupt Request:** Active low, open drain, output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
X1/CLK	X	X	X	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. When a crystal is used, a capacitor must be connected from this pin to ground (see Figure 5).
X2	X	X		I	**Crystal 2:** Connection for other side of the crystal. When a crystal is used, a capacitor must be connected from this pin to ground (see Figure 5). If X1/CLK is driven from an external source, this pin should be grounded.
RxDA	X	X	X	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.

PIN DESCRIPTION (Continued)

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
RxDB	X	X	X	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	X	X	X	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	X	X	X	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	X	X		O	**Output 0:** General purpose output, or channel A request to send (RTSAN, active low). Can be deactivated automatically on receive or transmit.
OP1	X	X		O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low). Can be deactivated automatically on receive or transmit.
OP2	X			O	**Output 2:** General purpose output, or channel A transmitter 1× or 16× clock output, or channel A receiver 1× clock output.
OP3	X			O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1× clock output, or channel B receiver 1× clock output.
OP4	X			O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.
OP5	X			O	**Output 5:** General purpose output, or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	X			O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output.
OP7	X			O	**Output 7:** General purpose output, or channel B open drain, active low, TxRDYB output.
IP0	X			I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN).
IP1	X			I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN).
IP2	X	X		I	**Input 2:** General purpose input, or counter/timer external clock input.
IP3	X			I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP4	X			I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	X			I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP6	X			I	**Input 6:** General purpose input or channel B receiver external clock input (RxCB). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
V_{CC}	X	X	X	I	**Power Supply:** +5V supply input
GND	X	X	X	I	**Ground**

BLOCK DIAGRAM


D0-D7
8
BUS BUFFER
RDN
WRN
CEN
A0-A3
4
RESET
OPERATION CONTROL
ADDRESS DECODE
R/W CONTROL
INTERRUPT CONTROL
IMR
ISR
INTRN
TIMING
BAUD RATE GENERATOR
CLOCK SELECTORS
COUNTER/ TIMER
XTAL OSC
CSRA
CSRB
ACR
CTUR
CTLR
X1/CLK
X2
CONTROL
TIMING
INTERNAL DATA BUS
CHANNEL A
TRANSMIT HOLDING REG
TRANSMIT SHIFT REGISTER
RECEIVE HOLDING REG (3)
RECEIVE SHIFT REG
MRA1,2
CRA
SRA
TxDA
RxDA
CHANNEL B (AS ABOVE)
TxDB
RxDB
INPUT PORT
CHANGE OF STATE DETECTORS (4)
IPCR
ACR
7
IP0-IP6
OUTPUT PORT
FUNCTION SELECT LOGIC
OPCR
OPR
8
OP0-OP7
V_{CC}
GND
BD00101S

Dual Asynchronous Receiver/Transmitter (DUART) SCN2681

BLOCK DIAGRAM

The 2681 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurrence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions.

Outputs OP3 – OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART.

If an external is used instead of a crystal, X1 should be driven using a configuration similar to the one in Figure 5. The input clock must be capable of attaining a V_{IH} of 4.4V.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4k baud. The clock outputs from the BRG are at 16× the actual baud rate. The counter/timer can be used as a timer to produce a 16× clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection, for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Communications Channels A And B

Each communications channel of the 2681 comprises a full-duplex asynchronous receiver/transmitter (UART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The input port pulse detection circuitry uses a 38.4kHz sampling clock derived from one of the baud rate generator taps. This results in a sampling period of slightly more than 25μs (this assumes that the clock input is 3.6864MHz). The detection circuitry, in order to guarantee that a true change in level has occurred, requires two successive samples at the new logic level be observed. As a consequence, the minimum duration of the signal change is 25μs if the transition occurs "coincident with the first sample pulse." The 50μs time refers to the situation in which the change of state is "just missed" and the first change of state is not detected until 25μs later.

The inputs to this unlatched 7-bit port can be read by the CPU by performing a read operation at address D16. A high input results in a logic 1 while a low input results in a logic 0. D7 will always be read as a logic 1. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs, lasting longer than 25 – 50μs, will set the corresponding bit in the input port change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E16 with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F16 with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The 2681 is conditioned to transmit data when the transmitter is enabled through the command register. The 2681 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted. If it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted; only if the transmitter has been disabled.

Receiver

The 2681 is conditioned to receive data when enabled through the command register. The receiver looks for a high-to-low (mark-to-space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16× clock for 7½ clocks (16× clock mode) or at the next rising edge of the bit time clock (1× clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped', thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical-OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected; the character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set-upon receipt of the start bit of the new (overrunning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to '11' for channels A and B, respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, with receivers that are normally disabled, examine the received data stream and 'wake-up' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in Table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems.

Table 1. 2681 Register Addressing

A3	A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	Rx Holding Register A (RHRA)	Tx Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	Rx Holding Register B (RHRB)	Tx Holding Register B (THRB)
1	1	0	0	*Reserved*	*Reserved*
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x, switches the pointer to MR2x. The pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to Table 2 for register bit descriptions. The reserved registers at addresses H'02' and H'0A' should never be read during normal operation since they are reserved for internal diagnostics.

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control

This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR 1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select

This bit selects either the channel A receiver ready status (RxRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the accumulation (logical-OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

Table 2. Register Bit Formats

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RxRTS CONTROL	RxINT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHARACTER	
MR1A **MR1B**	0 = no 1 = yes	0 = RxRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = multi-drop mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	CHANNEL MODE		TxRTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A **MR2B**	00 = Normal 01 = Auto-echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

NOTE:
*Add 0.5 to values shown for 0–7 if channel is programmed for 5 bits/char.

Table 2. Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
CSRA **CSRB**	See Text				See Text			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
		MISCELLANEOUS COMMANDS			DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
CRA **CRB**	Not used – must be 0	See Text			0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA **SRB**	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

NOTE:
*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits (4:0). These bits are cleared by a 'reset error status' command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1 = TxRDYB	0 = OPR[6] 1 = TxRDYA	0 = OPR[5] 1 = RxRDY/ FFULLB	0 = OPR[4] 1 = RxRDY/ FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB(1×) 11 = RxCB(1×)		00 = OPR[2] 01 = TxCA(16×) 10 = TxCA(1×) 11 = RxCA(1×)	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See Table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	DELTA IP3	DELTA IP2	DELTA IP1	DELTA IP0	IP3	IP2	IP1	IP0
IPCR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	INPUT PORT CHANGE	DELTA BREAK B	RxRDY/ FFULLB	TxRDYB	COUNTER READY	DELTA BREAK A	RxRDY/ FFULLA	TxRDYA
ISR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

Table 2. Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IMR	IN. PORT CHANGE INT	DELTA BREAK B INT	RxRDY/ FFULLB INT	TxRDYB INT	COUNTER READY INT	DELTA BREAK A INT	RxRDY/ FFULLA INT	TxRDYA INT
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTUR	C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTLR	C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

MR1A[4:3] — Channel A Parity Mode Select
If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select
This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits Per Character Select
This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select
Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission; i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is relocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control
This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR 2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically termi-

nate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Verify the message is sent by waiting until the transmit ready status (TxRDY) is asserted. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-send Control

If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select

This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1 9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1 1/16 to 2 stop bits can be programmed in increments of 1/16-bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1× clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select

This field selects the baud rate clock for the channel A receiver as follows:

	Baud Rate	
CSRA[7:4]	**ACR[7] = 0**	**ACR[7] = 1**
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4k	19.2k
1 1 0 1	Timer	Timer
1 1 1 0	IP4 — 16×	IP4 — 16×
1 1 1 1	IP4 — 1×	IP4 — 1×

The receiver clock is always a 16× clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select

This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRA[3:0]	**ACR[7] = 0**	**ACR[7] = 1**
1 1 1 0	IP3 — 16×	IP3 — 16×
0 1 1 1	IP3 — 1×	IP3 — 1×

The transmitter clock is always a 16× clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select

This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[7:4]	**ACR[7] = 0**	**ACR[7] = 1**
1 1 1 0	IP6 — 16×	IP6 — 16×
0 1 1 1	IP6 — 1×	IP6 — 1×

The receiver clock is always a 16× clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select

This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[3:4]	**ACR[7] = 0**	**ACR[7] = 1**
1 1 1 0	IP5 — 16×	IP5 — 16×
1 1 1 1	IP5 — 1×	IP5 — 1×

The transmitter clock is always a 16× clock except for CSRB[3:0] = 1111.

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands

The encoded value of this field may be used to specify a single command as follows:

CRA[6:4]	COMMAND
0 0 0	No command.
0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is reset.
0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.

CRA[6:4]	COMMAND
1 1 0	Start break. Forces the TxDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
1 1 1	Stop Break. The TxDA line will go high (marking) within two bit times. TxDA will remain high for one bit time before the next character, if any, is transmitted.

CRA[3] — Disable Channel A Transmitter
This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter
Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver
This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver
Enables operation of the channel A receiver. If not in the special wakeup mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break
This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received: further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1× clock).

When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error
This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error
This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error
This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set-upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA)
This bit will be set when the channel A transmitter underruns, i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA)
This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

SRA[1] — Channel A FIFO Full (FFULLA)
This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA)
This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, only if (after this read) there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select
This bit programs the OP7 output to provide one of the following:

–The complement of OPR[7]

–The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select
This bit programs the OP6 output to provide one of the following:

–The complement of OPR[6]

–The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select
This bit programs the OP5 output to provide one of the following:

–The complement of OPR[5]

-The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select
This bit programs the OP4 output to provide one of the following:

-The complement of OPR[4]

-The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select
This field programs the OP3 output to provide one of the following:

-The complement of OPR[3]

-The counter/timer output, in which case OP3 acts as an open-drain output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.

-The 1× clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1× clock is output.

-The 1× clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1× clock is output.

OPCR[1:0] — OP2 Output Select
This field programs the OP2 output to provide one of the following:

-The complement of OPR[2]

-The 16× clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be a 1× clock if CSRA[3:0] = 1111.

-The 1× clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1× clock is output.

-The 1× clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1× clock is output.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select
This bit selects one of two sets of baud rates to be generated by the BRG:

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05k, 1.2k, 2.4k, 4.8k, 7.2k, 9.6k, and 38.4k baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k, and 19.2k baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in Table 3.

ACR[6:4] — Counter/Timer Mode And Clock Source Select
This field selects the operating mode of the counter/timer and its clock source as shown in Table 4.

ACR[3:0] — IP3, IP2, IP1, IP0 Change Of State Interrupt Enable
This field selects which bits of the input port change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change-of-State
These bits are set when a change-of-state, as defined in the input port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State
These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initialized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status
This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

ISR[6] — Channel B Change In Break
This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B 'reset break change interrupt' command.

ISR[5] — Channel B Receiver Ready Or FIFO Full
The function of this bit is programmed by MR1B[6]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full; i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[4] — Channel B Transmitter Ready
This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready
In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change In Break
This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready Or FIFO Full
The function of this bit is programmed by MR1A[6]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be

**Table 3. Baud Rate Generator Characteristics
Crystal or Clock = 3.6864MHz**

NOMINAL RATE (BAUD)	ACTUAL 16× CLOCK (kHz)	ERROR (%)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2k	307.2	0
38.4k	614.4	0

NOTE:
Duty cycle of 16× clock is 50% ±1%

Table 4. ACR 6:4 Field Definition

ACR 6:4	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)
0 0 1	Counter	TxCA — 1× clock of channel A transmitter
0 1 0	Counter	TxCB — 1× clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)
1 0 1	Timer	External (IP2) divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready
This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3 – OP7 or the reading of the ISR.

CTUR And CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs, respectively, of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 0002_{16}. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A3 – A0 = 1110) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR.

The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A3 – A0 = 1111). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

On power up and after reset, the timer/counter runs in timer mode and can only be restarted. Because it cannot be shut off or stopped, and runs continuously in timer mode, it is recommended that at initialization, the output port (OP3) should be masked off through the OPCR[3:2] = 00 until the T/C is programmed to the desired operational state.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8 bits to the upper 8 bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

Dual Asynchronous Receiver/Transmitter (DUART) SCN2681

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	-65 to +150	°C
	All voltages with respect to ground[3]	-0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS Min	Typ	Max	UNIT
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2			V
V_{IH}	Input high voltage (X1/CLK)		4			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	-10		10	μA
I_{LL}	Data bus 3-State leakage current	$V_O = 0.4$ to V_{CC}	-10		10	μA
I_{X1L}	X1/CLK low input current	$V_{IN} = 0$, X2 grounded	-4	-2	0	mA
		$V_{IN} = 0$, X2 floated	-3	-1.5	0	mA
I_{X1H}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 grounded	-1	0.2	1	mA
		$V_{IN} = V_{CC}$, X2 floated	0	3.5	10	mA
I_{X2L}	X2 low input current	$V_{IN} = 0$, X1/CLK floated	-100	-30	0	μA
I_{X2H}	X2 high input current	$V_{IN} = V_{CC}$, X1/CLK floated	0	+30	100	μA
I_{OC}	Open-collector output leakage current	$V_O = 0.4$ to V_{CC}	-10		10	μA
I_{CC}	Power supply current				150	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all inputs except X1/CLK swing between 0.4V and 2.4V with a transition time of 20ns maximum. For X1/CLK this swing is between 0.4V and 4.4V. All time measurements are referenced at input voltages of 0.8V and 2V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. Test condition for outputs: $C_L = 150pF$, except interrupt outputs. Test condition for interrupt outputs: $C_L = 50pF$, $R_L = 2.7k\Omega$ to V_{CC}.
8. Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the 'strobing' input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN. CEN and RDN (also CEN and WRN) are AND'ed internally. As a consequence, the signal asserted last initiates the cycle and the signal negated first terminates the cycle.
9. If CEN is used as the 'strobing' input, the parameter defines the minimum high times between one CEN and the next. The RDN signal must be negated for t_{RWD} to guarantee that any status register changes are valid.
10. Consecutive write operations to the same command register require at least three edges of the X1 clock between writes.

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%[4, 5, 6, 7]

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Typ	Max	
Reset timing (Figure 1)					
t_{RES}	RESET pulse width	1.0			μs
Bus timing (Figure 2)[8]					
t_{AS}	A0 – A3 setup time to RDN, WRN low	10			ns
t_{AH}	A0 – A3 hold time from RDN, WRN high	0			ns
t_{CS}	CEN setup time to RDN, WRN low	0			ns
t_{CH}	CEN hold time from RDN, WRN high	0			ns
t_{RW}	WRN, RDN pulse width	225			ns
t_{DD}	Data valid after RDN low			175	ns
t_{DF}	Data bus floating after RDN high			100	ns
t_{DS}	Data setup time before WRN high	100			ns
t_{DH}	Data hold time after WRN high	20			ns
t_{RWD}	High time between READs and/or WRITEs[9,10]	200			ns
Port timing (Figure 3)[8]					
t_{PS}	Port input setup time before RDN low	0			ns
t_{PH}	Port input hold time after RDN high	0			ns
t_{PD}	Port output valid after WRN high			400	ns
Interrupt timing (Figure 4)					
t_{IR}	INTRN (or OP3 – OP7 when used as interrupts) negated from:				
	Read RHR (RxRDY/FFULL interrupt)			300	ns
	Write THR (TxRDY interrupt)			300	ns
	Reset command (delta break interrupt)			300	ns
	Stop C/T command (counter interrupt)			300	ns
	Read IPCR (input port change interrupt)			300	ns
	Write IMR (clear of interrupt mask bit)			300	ns
Clock timing (Figure 5)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2	3.6864	4	MHz
t_{CTC}	CTCLK (IP2) high or low time	100			ns
f_{CTC}	CTCLK (IP2) frequency	0		4	MHz
t_{RX}	RxC high or low time	220			ns
f_{RX}	RxC frequency (16×)	0		2	MHz
	(1×)	0		1	MHz
t_{TX}	TxC high or low time	220			ns
f_{TX}	TxC frequency (16×)	0		2	MHz
	(1×)	0		1	MHz
Transmitter timing (Figure 6)					
t_{TXD}	TxD output delay from TxC low			350	ns
t_{TCS}	Output delay from TxC low to TxD data output	0		150	ns
Receiver timing (Figure 7)					
t_{RXS}	RxD data setup time to RxC high	240			ns
t_{RXH}	RxD data hold time from RxC high	200			ns

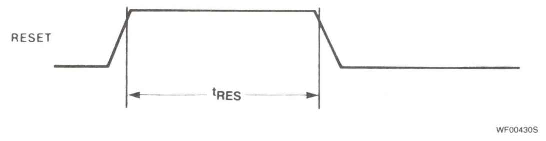

RESET
tRES
WF00430S

Figure 1. Reset Timing

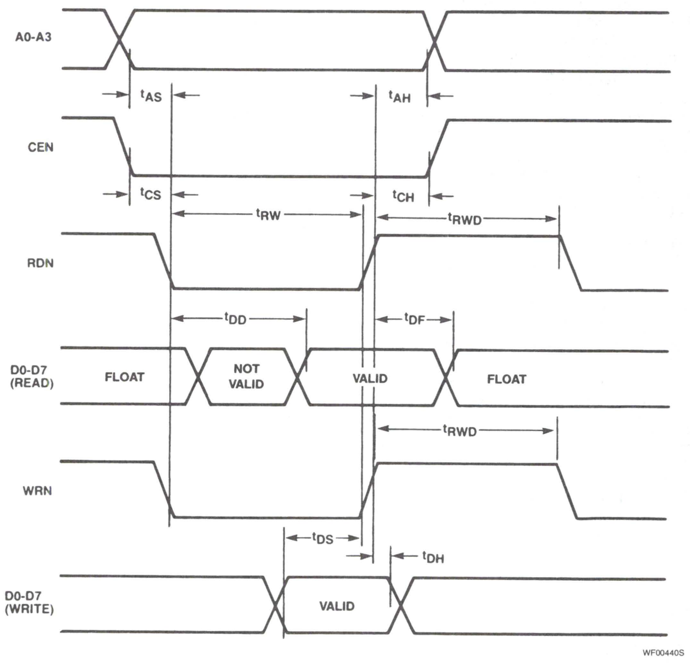

A0-A3
tAS
tAH
CEN
tCS
tCH
tRW
tRWD
RDN
tDD
tDF
D0-D7 (READ)
FLOAT
NOT VALID
VALID
FLOAT
tRWD
WRN
tDS
tDH
D0-D7 (WRITE)
VALID
WF00440S

Figure 2. Bus Timing

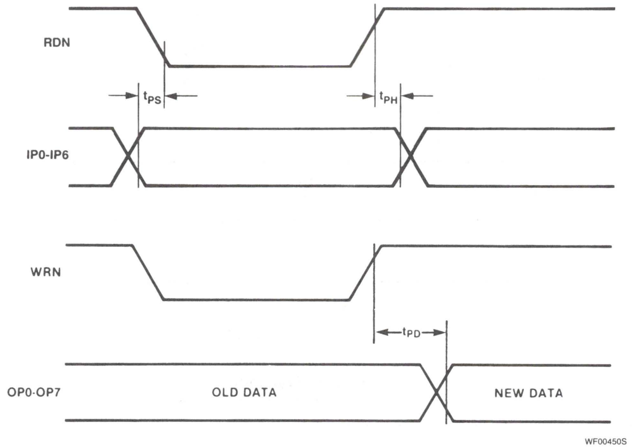
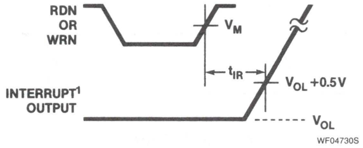

RDN
tPS
tPH
IP0-IP6
WRN
tPD
OP0-OP7
OLD DATA
NEW DATA
WF00450S

Figure 3. Port Timing


RDN
OR
WRN
VM
tIR
VOL +0.5V
INTERRUPT1
OUTPUT
VOL
WF04730S

NOTES:

1. INTRN or OP3 – OP7 when used as interrupt outputs.
2. The test for open drain outputs is intended to guarantee switching of the output transistor. Measurement of this response is referenced from the midpoint of the switching signal, V_M, to a point 0.5V above V_{OL}. This point represents noise margin that assures true switching has occurred. Beyond this level, the effects of external circuitry and test environment are pronounced and can greatly affect the resultant measurement.

Figure 4. Interrupt Timing

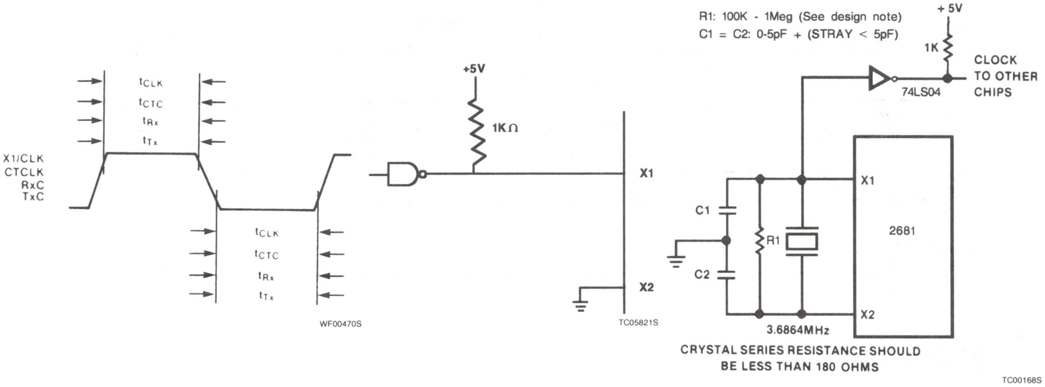

tCLK
tCTC
tRx
tTx
X1/CLK
CTCLK
RxC
TxC
WF00470S
+5V
1KΩ
X1
X2
TC05821S
R1: 100K - 1Meg (See design note)
C1 = C2: 0-5pF + (STRAY < 5pF)
+5V
1K
CLOCK TO OTHER CHIPS
74LS04
C1
C2
R1
X1
X2
2681
3.6864MHz
CRYSTAL SERIES RESISTANCE SHOULD BE LESS THAN 180 OHMS
TC00168S

Figure 5. Clock Timing

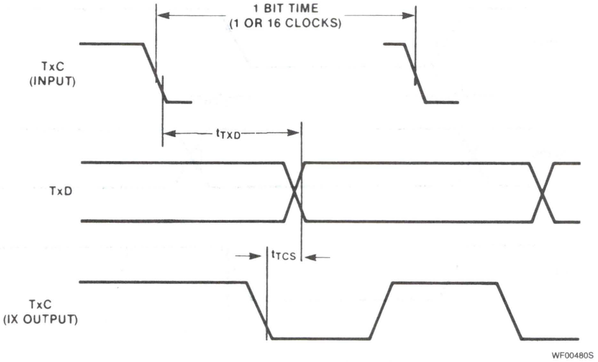

1 BIT TIME
(1 OR 16 CLOCKS)
TxC
(INPUT)
t_{TXD}
TxD
t_{TCS}
TxC
(IX OUTPUT)
WF00480S

Figure 6. Transmit

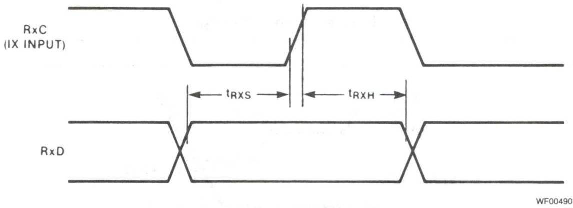
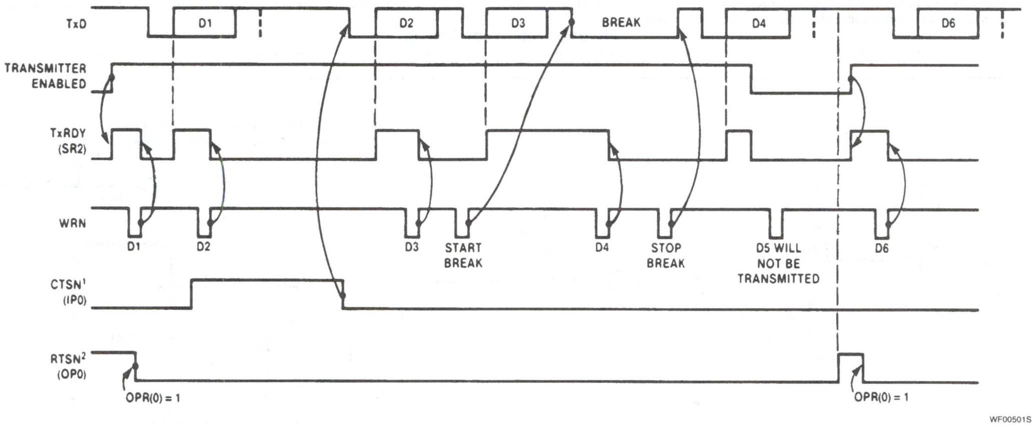
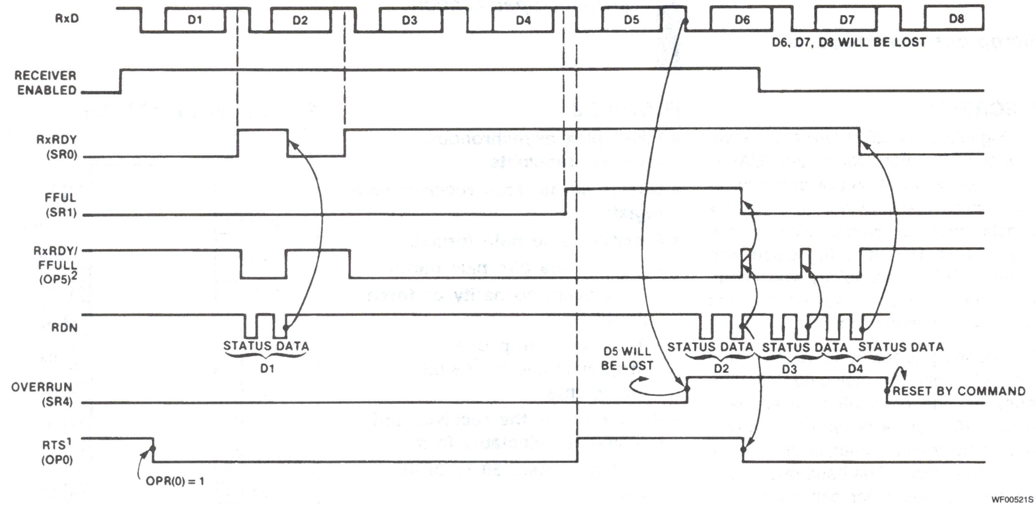

RxC
(IX INPUT)
t_{RXS}
t_{RXH}
RxD
WF00490S

Figure 7. Receive


TxD
D1
D2
D3
BREAK
D4
D6
TRANSMITTER
ENABLED
TxRDY
(SR2)
WRN
D1
D2
D3
START
BREAK
D4
STOP
BREAK
D5 WILL
NOT BE
TRANSMITTED
D6
CTSN¹
(IP0)
RTSN²
(OP0)
OPR(0) = 1
OPR(0) = 1
WF00501S

NOTES:
1. Timing shown for MR2(4) = 1.
2. Timing shown for MR2(5) = 1.

Figure 8. Transmitter Timing


RxD
D1
D2
D3
D4
D5
D6
D7
D8
D6, D7, D8 WILL BE LOST
RECEIVER ENABLED
RxRDY (SR0)
FFUL (SR1)
RxRDY/FFULL (OP5)2
RDN
STATUS DATA
D1
D5 WILL BE LOST
STATUS DATA
D2
STATUS DATA
D3
STATUS DATA
D4
OVERRUN (SR4)
RESET BY COMMAND
RTS1 (OP0)
OPR(0) = 1
WF00521S

NOTES:
1. Timing shown for MR1(7) = 1.
2. Shown for OPCR(4) = 1 and MR(6) = 0.

Figure 9. Receiver Timing

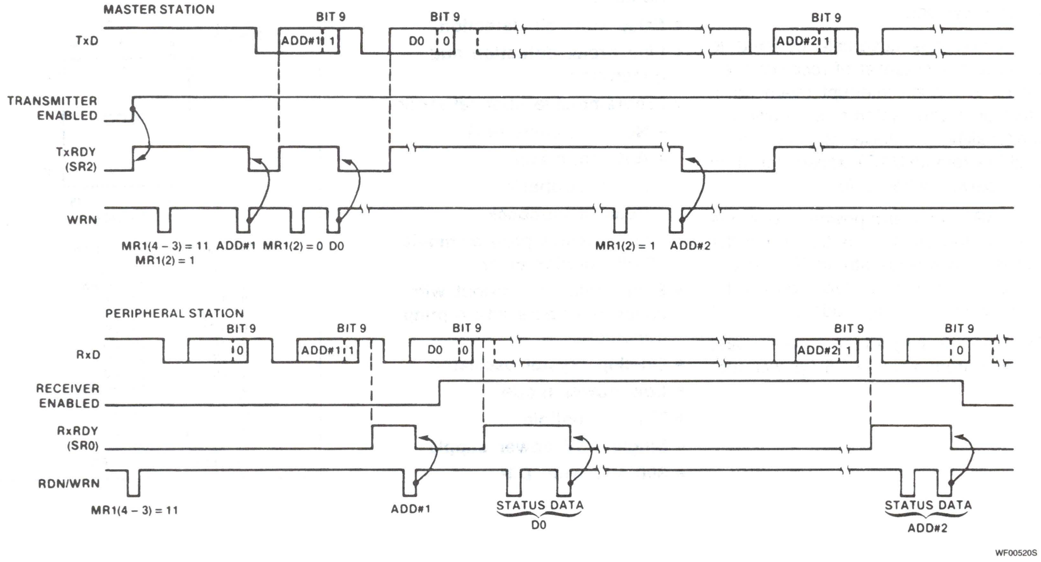

MASTER STATION
BIT 9
BIT 9
BIT 9
TxD
ADD#1 1
D0 0
ADD#2 1
TRANSMITTER ENABLED
TxRDY (SR2)
WRN
MR1(4 – 3) = 11
MR1(2) = 1
ADD#1
MR1(2) = 0
D0
MR1(2) = 1
ADD#2
PERIPHERAL STATION
BIT 9
BIT 9
BIT 9
BIT 9
BIT 9
RxD
0
ADD#1 1
D0 0
ADD#2 1
0
RECEIVER ENABLED
RxRDY (SR0)
RDN/WRN
MR1(4 – 3) = 11
ADD#1
STATUS DATA
D0
STATUS DATA
ADD#2
WF00520S

Figure 10. Wake-Up Mode

Signetics

Microprocessor Products

SCC2691
Universal Asynchronous Receiver/Transmitter (UART)

Product Specification

DESCRIPTION

The Signetics SCC2691 Universal Asynchronous Receiver/Transmitter (UART) is a single-chip CMOS-LSI communications device that provides a full-duplex asynchronous receiver/transmitter in a single 24-pin DIP. It is fabricated with Signetics CMOS technology which combines the benefits of high density and low power consumption.

The operating speed of the receiver and transmitter can be selected independently as one of eighteen fixed baud rates, a 16× clock derived from a programmable counter/timer, or an external 1× or 16× clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the UART particularly attractive for dual-speed channel applications such as clustered terminal systems.

The receiver is quadruple buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a handshaking capability is provided to disable a remote UART transmitter when the receiver buffer is full.

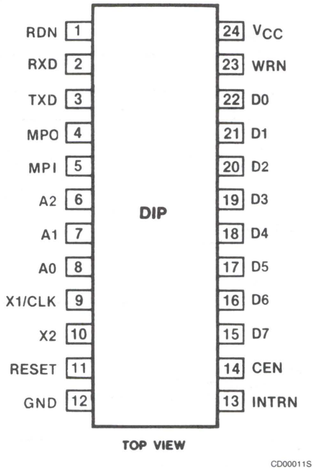
The UART provides a power-down mode in which the oscillator is frozen but the register contents are stored. This results in reduced power consumption on the order of several magnitudes.

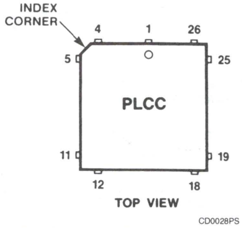
The UART is fully TTL compatible and operates from a single +5V power supply.

FEATURES

- **Full-duplex asynchronous receiver/transmitter**
- **Quadruple buffered receiver data register**
- **Programmable data format:**
  - **5 to 8 data bits plus parity**
  - **Odd, even, no parity or force parity**
  - **1, 1.5 or 2 stop bits programmable in 1/16-bit increments**
- **Baud rate for the receiver and transmitter selectable from:**
  - **18 fixed rates: 50 to 38.4k baud**
  - **One user-defined rate derived from programmable timer/counter**
  - **External 1× or 16× clock**
- **Parity, framing, and overrun error detection**
- **False start bit detection**
- **Line break detection and generation**
- **Programmable channel mode**
  - **Normal (full-duplex)**
  - **Automatic echo**
  - **Local loopback**
  - **Remote loopback**
- **Multi-function programmable 16-bit counter/timer**
- **Single interrupt output with seven maskable interrupting conditions**
- **On-chip crystal oscillator**
- **Low power mode**
- **TTL compatible**
- **Single +5V power supply**
- **300 MIL wide DIP**

PIN CONFIGURATIONS





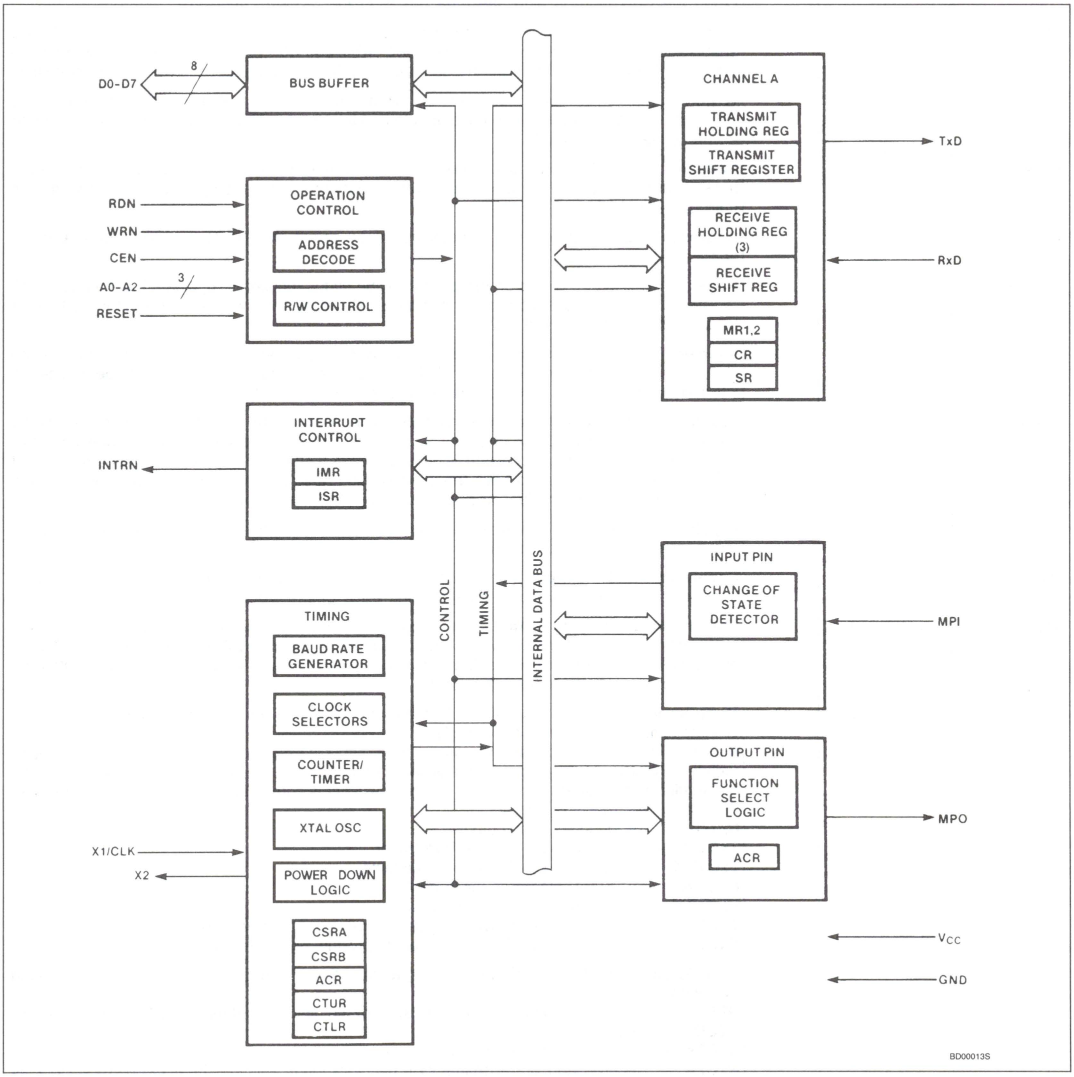
Pin	Function	Pin	Function
1	V_{DD}	15	V_{SS}
2	RDN	16	INTRN
3	RXD	17	CEN
4	TXD	18	D7
5	MPO	19	D6
6	MPI	20	D5
7	NC	21	D4
8	NC	22	D3
9	A2	23	NC
10	A1	24	D2
11	A0	25	D1
12	X1/CLK	26	NC
13	X2	27	D0
14	RESET	28	WRN

Universal Asynchronous Receiver/Transmitter (UART) SCC2691

ORDERING INFORMATION

PACKAGES	$V_{CC} = +5V \pm 10\%$, $T_A = 0°C$ to $+70°C$
Plastic DIP	SCC2691AC1N24
Plastic LCC	SCC2691AC1A28

BLOCK DIAGRAM


D0-D7
8
BUS BUFFER
CHANNEL A
TRANSMIT HOLDING REG
TRANSMIT SHIFT REGISTER
TxD
RDN
WRN
CEN
A0-A2
3
RESET
OPERATION CONTROL
ADDRESS DECODE
R/W CONTROL
RECEIVE HOLDING REG (3)
RECEIVE SHIFT REG
RxD
MR1,2
CR
SR
INTERRUPT CONTROL
INTRN
IMR
ISR
CONTROL
TIMING
INTERNAL DATA BUS
INPUT PIN
CHANGE OF STATE DETECTOR
MPI
TIMING
BAUD RATE GENERATOR
CLOCK SELECTORS
COUNTER/ TIMER
XTAL OSC
POWER DOWN LOGIC
CSRA
CSRB
ACR
CTUR
CTLR
X1/CLK
X2
OUTPUT PIN
FUNCTION SELECT LOGIC
ACR
MPO
VCC
GND
BD00013S

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
D0 – D7	22 – 15	26 – 24 22 – 18	I/O	**Data Bus:** Active high 8-bit bidirectional 3-State data bus. Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the UART take place over this bus. The direction of the transfer is controlled by the WRN and RDN inputs when the CEN input is low. When the CEN input is high, the data bus is in the 3-State condition.
CEN	14	17	I	**Chip Enable:** Active low input. When low, data transfers between the CPU and the UART are enabled on D0 – D7 as controlled by the WRN, RDN, and A0 – A2 inputs. When CEN is high, the UART is effectively isolated from the data bus and D0 – D7 are placed in the 3-State condition.
WRN	23	27	I	**Write Strobe:** Active low input. A low on this pin while CEN is low causes the contents of the data bus to be transferred to the register selected by A0 – A2. The transfer occurs on the trailing (rising) edge of the signal.
RDN	1	2	I	**Read Strobe:** Active low input. A low on this pin while CEN is low causes the contents of the register selected by A0 – A2 to be placed on the data bus. The read cycle begins on the leading (falling) edge of RDN.
A0 – A2	8 – 6	10 – 8	I	**Address Inputs:** Active high address inputs to select the UART registers for read/write operations.
RESET	11	13	I	**Reset:** Master reset. A high on this pin clears the status register (SR), clears the interrupt mask register (IMR), and places the receiver and transmitter in the inactive state causing the TxD output to go to the marking (high) state.
INTRN	13	16	O	**Interrupt Request:** This active low output is asserted upon occurrence of one or more of seven maskable interrupting conditions. The CPU can read the interrupt status register to determine the interrupting condition(s). This open-drain output requires a pull-up resistor.
X1/CLK	9	11	I	**Crystal 1:** Crystal or external clock input. When using the crystal oscillator, this pin serves as the connection for one side of the crystal. If a crystal is not used, an external clock is supplied at this input. An external clock (or crystal) is required even if the internal baud rate generator is not utilized. This clock is used to drive the internal baud rate generator, as an optional input to the timer/counter, and to provide other clocking signals required by the chip.
X2	10	12	O	**Crystal 2:** Connection for other side of crystal. If an external source is used instead of a crystal, this connection should be open.
RxD	2	3	I	**Receiver Serial Data Input:** The least significant bit is received first. If external receiver clock is specified, this input is sampled on the rising edge of the clock.
TxD	3	4	O	**Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the marking (high) condition when the transmitter is idle or disabled and when the UART is operating in local loopback mode. If external transmitter clock is specified, the data is shifted on the falling edge of the transmitter clock.
MPO	4	5	O	**Multi-Purpose Output:** One of the following functions can be selected for this output pin by programming the auxiliary control register: **RTSN** – Request to send active low output. This output is asserted and negated via the command register. By appropriate programming of the mode registers, RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted or when the receiver FIFO and shift register are full. **C/TO** — The counter/timer output. **TxC1×** — The 1× clock for the transmitter. **TxC16×** — The 16× clock for the transmitter. **RxC1×** — The 1× clock for the receiver. **RxC16×** — The 16× clock for the receiver. **TxRDY** — The transmitter holding register empty signal. Active low interrupt. **RxRDY/FFULL** — The receiver FIFO not empty/full signal. Active low interrupt.
MPI	5	6	I	**Multi-Purpose Input:** This pin can be programmed to serve as an input for one of the following functions: **GPI** — General purpose input. The current state of the pin can be determined by reading the ISR. **CTSN** — Clear-to-Send active low input. **CTCLK** — Counter/timer external clock input. **RTCLK** — Receiver and/or transmitter external clock input. This may be a 1× or 16× clock as programmed by CSR[3:0] or CSR[7:4].
V_{CC}	24	28	I	**Power Supply:** +5V supply input.
GND	12	14	I	**Ground**

BLOCK DIAGRAM

As shown on the block diagram, the UART consists of: data bus buffer, interrupt control, operation control, timing, receiver and transmitter.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data buses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the UART.

Interrupt Control

A single interrupt output (INTRN) is provided which is asserted upon the occurrence of any of the following internal events:

- Transmit holding register ready
- Transmit shift register empty
- Receive holding register ready or FIFO full
- Change in break received status
- Counter reached terminal count
- Change in MPI input

Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR can be programmed to select only certain of the above conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. However, the bits of the ISR are not masked by the IMR.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are shown in Table 1.

Table 1. Register Addressing

A2	A1	A0	READ (RDN=0)	WRITE (WRN=0)
0	0	0	MR1,MR2	MR1, MR2
0	0	1	SR	CSR
0	1	0	Reserved*	CR
0	1	1	RHR	THR
1	0	0	Reserved*	ACR
1	0	1	ISR	IMR
1	1	0	CTU	CTUR
1	1	1	(CTL)	CTLR

NOTE:

*Reserved registers should never be read during normal operation since they are reserved for internal diagnostics.

ACR = Auxiliary control register
CR = Command register
CSR = Clock select register
CTL = Counter/timer lower
CTLR = Counter/timer lower register
CTU = Counter/timer upper
CTUR = Counter/timer upper register
MR = Mode register A
SR = Status register
THR = Tx holding register

Mode registers 1 and 2 are accessed via an auxiliary pointer. The pointer is set to MR1 by RESET or by issuing a reset pointer command via the command register. Any read or write of the mode register while the pointer is at MR1 switches the pointer to MR2. The pointer then remains at MR2 so that subsequent accesses are to MR2, unless the pointer is reset to MR1 as described above.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and two clock selectors.

The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs with a minimum of external components. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. If an external clock is used instead of a crystal, X1/CLK is driven using a configuration similar to the one in Figure 5. However, the input high voltage must be capable of attaining 4.4V. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock frequency, within the limits specified in the electrical specifications, must be supplied even if the internal BRG is not used.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4K baud. Thirteen of these are available simultaneously for use by the receiver and transmitter. Eight are fixed, and one of two sets of five can be selected by programming ACR[7]. The clock outputs from the BRG are at 16× the actual baud rate. The counter/timer can be used as a timer to produce a 16× clock for any other baud rate by counting down the crystal clock or an external clock. The clock selectors allow the independent selection by the receiver and transmitter of any of these baud rates or an external timing signal.

The C/T operation is programmed by ACR[6:4]. One of eight timing sources can be used as the input to the C/T. The output of the C/T is available to the clock selectors and can also be programmed by ACR[2:0], to be output on the MPO pin.

In the timer mode, the C/T generates a square wave whose period is twice the number of clock periods loaded into the C/T upper and lower registers. The counter ready bit in the ISR is set once each cycle of the square wave. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half-periods will be affected. In this mode the C/T runs continuously and does not recognize the stop counter command (the command only resets the counter ready bit in the ISR). Receipt of a start C/T command causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR. Counting begins upon receipt of a start C/T command. Upon reaching terminal count, the counter ready bit in the ISR is set. The counter continues counting past the terminal count until stopped by the CPU. If MPO is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and the counter ready bit is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command following a stop counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8 bits to the upper 8 bits occurs between the times that both halves of the counter are read. However a subsequent start counter command causes the counter to begin a new count cycle using the values in CTUR and CTLR.

Receiver and Transmitter

The UART is a full-duplex asynchronous receiver/transmitter. The operating frequency for the receiver and transmitter can be selected independently from the baud rate generator, the counter/timer, or from an external input. Registers associated with the communications channel are the mode registers (MR1 and MR2), the clock select register (CSR), the command register (CR), the status register (SR), the transmit holding register (THR), and the receive holding register (RHR).

Transmitter

The transmitter accepts parallel data from the CPU and converts it to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the SR will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. In the 16× clock mode, this also resynchronizes the internal 1× transmitter clock so that transmission of the new character begins with minimum delay.

The transmitter can be forced to send a break (continuous low condition) by issuing a start break command via the CR. The break is terminated by a stop break command.

If the transmitter is disabled, it continues operating until the character currently being transmitted and the character in the THR, if any, are completely sent out. Characters cannot be loaded into the THR while the transmitter is disabled.

Receiver

The receiver accepts serial data on the RxD pin, converts the serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition, and presents the assembled character to the CPU. The receiver looks for a high-to-low (mark-to-space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled again each 16× clock for $7^1/2$ clocks (16× clock mode) or at the next rising edge of the bit time clock (1× clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the RHR and the RxRDY bit in the SR is set to a 1. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (i.e. framing error) and RxD remains low for one-half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled). The parity error, framing error and overrun error (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set.

If a break condition is detected (RxD is low for the entire character including the stop bit) only one character consisting of all zeros will be loaded into the FIFO and the received break bit in the SR is set to 1. The RxD input must return to a high condition for two successive clock edges of the 1× clock (internal or external) before a search for the next start bit begins.

RECEIVER FIFO

The RHR consists of a first-in-first-out (FIFO) queue with a capacity of three characters. Data is loaded from the receive shift register into the top-most empty position of the FIFO. The RXRDY bit in the status register (SR) is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three queue positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits are 'popped' thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are appended to each data character in the FIFO. Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the character mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the block mode, the status provided in the SR for these three bits is the logical-OR of the status for all characters coming to the top of the FIFO since the last reset error command was issued. In either mode, reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore, the SR should be read prior to reading the corresponding data character.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exists, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set upon receipt of the start bit of the new (overrunning) character.

WAKE-UP MODE

In addition to the normal transmitter and receiver operation described above, the UART incorporates a special mode which provides automatic wake-up of the receiver through address frame recognition for multiprocessor communications. This mode is selected by programming bits MR1[4:3] to '11'.

In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, whose receivers are normally disabled, examine the received data stream and 'wake-up' the CPU (by setting RxRDY) only upon receipt of an adress character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1[2]: MR1[2] = 0 transmits a zero in the A/D bit position which identifies the corresponding data bits as data, while MR1[2] = 1 transmits a one in the A/D bit position which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

While in this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one, but discards the received character if the received A/D bit is a zero. If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SR[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

MULTI-PURPOSE INPUT PIN

The MPI pin can be programmed as an input to one of several UART circuits. The function of the pin is selected by programming the appropriate control register (MR2[4], ACR[6:4], CSR[7:4, 3:0]). Only one of the functions may be selected at any given time. If CTS or GPI is selected, a change of state detector provided with the pin is activated. A high-to-low or low-to-high transition of the inputs lasting longer than 25 – 50μs sets the MPI change-of-state bit in the interrupt status register. The bit is cleared via a command. The change-of-state can be programmed to generate an interrupt to the CPU by setting the corresponding bit in the interrupt mask register.

The input port pulse detection circuitry uses a 38.4kHz sampling clock derived from one of the baud rate generator taps. This produces a sampling period of slightly more than 25μs (assuming a 3.6864MHz oscillator input). The detection circuitry, in order to guarantee that a true change in level has occurred, requires two successive samples at the new logic level be observed. As a consequence, the minimum duration of the signal change is 25μs if the transition occurs coincident with the first sample pulse. The 50μs time refers to the condition where the change of state is just missed and the first change of state is not detected until after an additional 25μs.

MULTI-PURPOSE OUTPUT PIN

This pin can be programmed to serve as a request-to-send output, the counter/timer output, the output for the 1× or 16× transmitter or receiver clocks, the TxRDY output or the RxRDY/FFULL output (see ACR [2:0] — MPO Output Select).

REGISTERS

The operation of the UART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. Addressing of the registers is as described in Table 1.

The contents of certain control registers are initialized to zero on RESET (see Reset pin description). Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems — e.g., changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. The contents of the MR, the CSR, and the ACR should only be changed while the receiver(s) and transmitter(s) are disabled, and certain changes to the ACR should only be made while the C/T is stopped.

The bit formats of the UART registers are depicted in Table 2.

MR1 — Mode Register 1

MR1 is accessed when the MR pointer points to MR1. The pointer is set to MR1 by RESET or by a set pointer command applied via CR. After reading or writing MR1, the pointers are set at MR2.

MR1[7] — Receiver Request-to-Send Control

This bit controls the deactivation of the RTSN output (MPO) by the receiver. This output is manually asserted and negated by commands applied via the command register. MR1[7] = 1 causes RTSN to be automatically negated upon receipt of a valid start bit if the receiver FIFO is full. RTSN is reasserted when an empty FIFO position is available. This feature can be used to prevent overrun in the receiver by using the RTSN output signal to control the CTS input of the transmitting device.

MR1[6] — Receiver Interrupt Select

This bit selects either the receiver ready status (RxRDY) or the FIFO full status (FFULL) to be used for CPU interrupts.

MR1[5] — Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break). In the character mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the block mode, the status provided in the SR for these bits is the accumulation (logical-OR) of the status for all characters coming to the top of the FIFO since the last reset error command was issued.

MR1[4:3] — Parity Mode Select

If with parity or force parity is selected , a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1[4:3] = 11 selects the channel to operate in the special wake-up mode.

MR1[2] — Parity Type Select

This bit selects the parity type (odd or even) if the with parity mode is programmed by MR1[4:3], and the polarity of the forced parity bit if the force parity mode is programmed. It has no effect if the no parity mode is programmed. In the special wake up mode, it selects the polarity of the A/D bit.

MR1[1:0] — Bits per Character Select

This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2 — Mode Register 2

MR2 is accessed when the channel MR pointer points to MR2, which occurs after any access to MR1. Accesses to MR2 do not change the pointer.

MR2[7:6] — Mode Select

The UART can operate in one of four modes: MR2[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU-to-receiver communication continues normally, but the CPU-to-transmitter link is disabled.

Two diagnostic modes can also be selected. MR2[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., the transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

When switching in and out of the various modes, the selected mode is activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of auto echo or remote loopback modes; if the deselection occurs just after the receiver has sampled the stop bit (indicated to be in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in auto echo mode until one full stop bit has been retransmitted.

MR2[5] — Transmitter Request-to-Send Control
This bit controls the deactivation of the RTSN output (MPO) by the transmitter. This output is manually asserted and negated by appropriate commands issued via the command register. MR2[5] = 1 causes RTSN to be reset automatically one bit time after the characters in the transmit shift register and in the THR (if any) are completely transmitted; includes the programmed number of stop bits if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2[5] = 1.
2. Enable transmitter.
3. Assert RTSN via command.
4. Send message.
5. Verify the next to last character of the message is being sent by waiting until transmitter ready is asserted. Disable transmitter after the last character is loaded into the THR.
6. The last character will be transmitted and RTSN will be reset one bit time after the last stop bit.

MR2[4] — Clear-to-Send Control
The state of this bit determines if the CTSN input (MPI) controls the operation of the transmitter. If this bit is 0, CTSN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSN each time it is ready to send a character. If it is asserted (low), the character is transmitted. If it is negated (high), the TxD output remains in the marking state and the transmission is delayed until CTSN goes low. Changes in CTSN while a character is being transmitted do not affect the transmission of that character. This feature can be used to prevent overrun of a remote receiver.

MR2[3:0] — Stop Bit Length Select
This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of $^{9}/_{16}$ to 1 and $1^{9}/_{16}$ to 2 bits, in increments of $^{1}/_{16}$ bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, $1^{1}/_{16}$ to 2 stop bits can be programmed in increments of $^{1}/_{16}$ bit. In all cases, the receiver only checks for a mark condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled). If an external 1× clock is used for the transmitter, MR2[3] = 0 selects one stop bit and MR2[3] = 1 selects two stop bits to be transmitted.

CSR — Clock Select Register

CSR[7:4] — Receiver Clock Select
When using a 3.6864MHz crystal or external clock input, this field selects the baud rate clock for the receiver as shown in Table 3.

Table 2. Register Bit Formats

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RxRTS CONTROL	RxINT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHAR	
MR1	0=no 1=yes	0=RXRDY 1=FFULL	0=char 1=block	00=with parity 01=force parity 10=no parity 11=special mode		0=even 1=odd	00=5 01=6 10=7 11=8	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	CHANNEL MODE		TxRTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2	00=Normal 01=Auto echo 10=Local loop 11=Remote loop		0=no 1=yes	0=no 1=yes	0=0.563 1=0.625 2=0.688 3=0.750	4=0.813 5=0.875 6=0.938 7=1.000	8=1.563 9=1.625 A=1.688 B=1.750	C=1.813 D=1.875 E=1.938 F=2.000

NOTE:
*Add 0.5 to values shown for 0 – 7, if channel is programmed for 5 bits/char.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
CSR	See Text				See Text			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	MISCELLANEOUS COMMANDS				DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
CR	See Text				0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes

Table 2. Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SR	**RECEIVED BREAK**	**FRAMING ERROR**	**PARITY ERROR**	**OVERRUN ERROR**	**TxEMT**	**TxRDY**	**FFULL**	**RxRDY**
	0=no 1=yes *	0=no 1=yes *	0=no 1=yes *	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes

NOTE:
*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits [7:5] from the top of the FIFO together with bits [4:0]. These bits are cleared by a reset error status command. In character mode they are reset when the corresponding data character is read from the FIFO.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ACR	**BRG SET SELECT**	**COUNTER/TIMER MODE AND SOURCE**			**POWER-DOWN MODE**	**MPO PIN FUNCTION SELECT**		
	0=set1 1=set2	See Text			0 = on 1 = off	000=RTSN 001=C/TO 010=TxC (1×) 011=TxC (16×)	100=RxC (1×) 101=RxC (16×) 110=TxRDY 111=RxRDY/FFULL	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ISR	**MPI PIN CHANGE**	**MPI PIN CURRENT STATE**		**COUNTER READY**	**DELTA BREAK**	**RxRDY/ FFULL**	**TxEMT**	**TxRDY**
	0=no 1=yes	0=low 1=high	not used	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IMR	**MPI CHANGE INT**	**MPI LEVEL INT**		**COUNTER READY INT**	**DELTA BREAK INT**	**RxRDY/ FFULL INT**	**TxEMT INT**	**TxRDY INT**
	0=off 1=on	0=off 1=on	not used	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTUR	**C/T[15]**	**C/T[14]**	**C/T[13]**	**C/T[12]**	**C/T[11]**	**C/T[10]**	**C/T[9]**	**C/T[8]**

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTLR	**C/T[7]**	**C/T[6]**	**C/T[5]**	**C/T[4]**	**C/T[3]**	**C/T[2]**	**C/T[1]**	**C/T[0]**

Table 3. Baud Rate

CSR[3:0]/ [7:4]	ACR[7]=0	ACR[7]=1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4k	19.2k
1 1 0 1	Timer	Timer
1 1 1 0	MPI — 16×	MPI — 16×
1 1 1 1	MPI — 1×	MPI — 1×

The receiver clock is always a 16× clock, except for CSR[7:4]=1111.

CSR[3:0] — Transmitter Clock Select
This field selects the baud rate clock for the transmitter. The field definition is as shown in Table 3.

CR — Command Register

CR is used to write commands to the UART. Multiple commands can be specified in a single write to CR as long as the commands are non-conflicting, e.g., the enable transmitter and reset transmitter commands cannot be specified in a single command word.

CR[7:4] — Miscellaneous Commands
The encoded value of this field may be used to specify a single command as follows:

0000 No command.

0001 Reset MR pointer. Causes the MR pointer to point to MR1.

0010 Reset receiver. Resets the receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.

0011 Reset transmitter. Resets the transmitter as if a hardware reset had been applied.

0100 Reset error status. Clears the received break, parity error, framing error, and overrun error bits in the status register (SR[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared), and in block mode to clear all error status after a block of data has been received.

0101 Reset break change interrupt. Causes the break detect change bit in the interrupt status register (ISR[3]) to be cleared to zero.

0110 Start break. Forces the TxD output low (spacing). If the transmitter is empty, the start of the break condition will be delayed up to two bit times. If the transmitter is active, the break begins when transmission of the character is completed. If a character is in the THR, the start of break is delayed until that character or any others loaded after it has been transmitted (TxEMT must be true before break begins). The transmitter must be enabled to start a break.

0111 Stop break. The TxD line will go high (marking) within two bit times. TxD will remain high for one bit time before the next character, if any, is transmitted.

1000 Start C/T. In counter or timer modes, causes the contents of CTUR/ CTLR to be preset into the counter/timer and starts the counting cycle. In timer mode, any counting cycle in progress when the command is issued is terminated. In counter mode, has no effect unless a stop C/T command was issued previously.

1001 Stop counter. In counter mode, stops operation of the counter/timer, resets the counter ready bit in the ISR, and forces the MPO output high if it is programmed to be the output of the C/T. In timer mode, resets the counter ready bit in the ISR but has no affect on the counter/timer itself or on the MPO output.

1010 Assert RTSN. Causes the RTSN output to be asserted (low).

1011 Negate RTSN. Causes the RTSN output to be negated (high).

1100 Reset MPI change interrupt. Causes the MPI change bit in the interrupt status register (ISR[7]) to be cleared to zero.

1101 Reserved.

111x Reserved.

CR[3] — Disable Transmitter
This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CR[2] — Enable Transmitter
Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CR[1] — Disable Receiver
This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special wake-up mode is programmed, the receiver operates even if it is disabled (see Wake-up Mode).

CR[0] — Enable Receiver
Enables operation of the receiver. If not in the special wake-up mode, this also forces the receiver into the search for start bit state.

SR — Channel Status Register

SR[7] — Received Break
This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received; further entries to the FIFO are inhibited until the RxD line returns to the marking state for at least one half bit time (two successive edges of the internal or external 1× clock).

When this bit is set, the change in break bit in the ISR (ISR[3]) is set. ISR[3] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry is capable of detecting breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must last until the end of the next character time in order for it to be detected.

SR[6] — Framing Error (FE)
This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SR[5] — Parity Error (PE)
This bit is set when the with parity or force parity mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special wake-up mode, the parity error bit stores the received A/D bit.

SR[4] — Overrun Error (OE)
This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a reset error status command.

SR[3] — Transmitter Empty (TxEMT)
This bit will be set when the transmitter underruns, i.e., both the transmit holding register (THR) and the transmit shift register are empty. However, this bit is not set until one character has been transmitted. It is set after transmission of the last stop bit of a

character, if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU, or when the transmitter is disabled.

SR[2] — Transmitter Ready (TxRDY)

This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, e.g., characters loaded in the THR while the transmitter is disabled will not be transmitted.

SR[1] — FIFO Full (FFULL)

This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the FIFO and there is no character in the receive shift register. If a character is waiting in the receive shift register because the FIFO is full, FFULL will be reset by the CPU read and then set by the transfer of the character to the FIFO, which causes all three FIFO positions to be occupied.

SR[0] — Receiver Ready (RxRDY)

This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, and no more characters are in the FIFO.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select

This bit selects one of two sets of baud rates generated by the BRG.

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05k, 1.2k, 2.4k, 4.8k, 7.2k, 9.6k, and 38.4k baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k, and 19.2k baud.

The selected set of rates is available for use by the receiver and transmitter.

ACR[6:4] — Counter/Timer Mode and Clock Source Select

This field selects the operating mode of the counter/timer and its clock source as follows:

ACR[6:4]	Mode	Clock Source
0 0 0	Counter	MPI pin
0 0 1	Counter	MPI pin divided by 16
0 1 0	Counter	TxC — 1× clock of the transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	MPI pin
1 0 1	Timer	MPI pin divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

ACR[3] — Power-Down Mode Select

This bit, when set to zero, selects the power-down mode. In this mode, the 2691 oscillator is stopped and all functions requiring this clock are suspended. The contents of all registers are saved. It is recommended that the transmitter and receiver be disabled prior to placing the 2691 in this mode. Note that this bit must be set to a logic 1 after power-up.

ACR[2:0] — MPO Output Select

This field programs the MPO output pin to provide one of the following:

000 Request-to-send active low output (RTSN). This output is asserted and negated via the command register. Mode RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted out or when the receiver FIFO and receiver shift register are full using MR2[5] and MR1[7], respectively.

001 The counter/timer output. In the timer mode, this output is a square wave with a period of twice the value (in clock periods) of the contents of the CTUR and CTLR. In the counter mode, the output remains high until the terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command.

010 The 1× clock for the transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a non-synchronized 1× clock is output.

011 The 16× clock for the transmitter. This is the clock selected by CSR[3:0], and is a 1× clock if CSR[3:0] = 1111.

100 The 1× clock for the receiver, which is the clock that samples the received data. If data is not being received, a non-synchronized 1× clock is output.

101 The 16× clock for the receiver. This is the clock selected by CSR[7:4], and is a 1× clock if CSR[7:4] = 1111.

110 The transmitter register empty signal, which is the complement of SR[2]. Active low output.

111 The receiver ready or FIFO full signal (complement of ISR[2]). Active low output.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output is asserted (low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no affect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status is provided regardless of the contents of the IMR.

ISR[7] — MPI Change-of-State

This bit is set when a change-of-state occurs at the MPI input pin. It is reset by a reset MPI change interrupt command.

ISR[6] — MPI Current State

This bit provides the current state of the MPI pin. The information is unlatched and reflects the state of the pin at the time the ISR is read.

ISR[4] — Counter Ready

In the counter mode of operation, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command. It is initialized to '0' when the chip is reset.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time the C/T reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the C/T.

ISR[3] — Change in Break

This bit, when set, indicates that the receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a reset break change interrupt command.

ISR[2] — Receiver Ready or FIFO Full

The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the receiver FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when FIFO is read and there is no character in the receiver shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

ISR[1] — Transmitter Empty
This bit is a duplicate of TxEMT (SR[3]).

ISR[0] — Transmitter Ready
This bit is a duplicate of TxRDY (SR[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is a '1', the INTRN output is asserted (low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask reading of the ISR.

CTUR AND CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs, respectively, the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded is 0002_{16}.

In the timer (programmable divider) mode, the C/T generates a square wave whose period is twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half-periods will be.

The counter ready status bit (ISR[4]) is set once each cycle of the square wave. The bit is reset by a stop counter command. The command, however, does not stop the C/T. The generated square wave is output on MPO if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR. Counting begins upon receipt of a start C/T command. Upon reaching the terminal count, the counter ready interrupt bit (ISR[4]) is set. The counter continues counting past the terminal count until stopped by the CPU. If MPO is programmed to be the output of the C/T, the output remains high until the terminal count is reached, at which time it goes low.

The output returns to the high state and ISR[4] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
V_{CC}	Voltage from V_{CC} to GND[3]	−0.5 to +7.0	V
V_S	Voltage from any pin to ground[3]	−0.5 to V_{CC} ±10%	V
P_D	Power dissipation	21	mW

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage					
	All except X1/CLK		2			V
	X1/CLK		$0.9V_{CC}$		V_{CC}	V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}[7]	Output high voltage					
	(except open drain outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-State leakage current	$V_O = 0.4$ to V_{CC}	−10		10	μA
I_{OD}	Open drain output leakage current	$V_O = 0.4$ to V_{CC}	−10		10	μA
I_{X1L}	X1/CLK low input current	$V_{IN} = 0$, X2 floated	−100	−30	0	μA
I_{X1H}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 floated	0	+30	100	μA
When oscillator is in power-down mode:						
I_{X1H}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 floated	2	6	10	mA
I_{X2L}	X2 low output current	$V_{OUT} = 0$, X1/CLK = V_{CC}			100	μA
I_{X2H}	X2 high output current	$V_{OUT} = V_{CC}$, X1/CLK = 0V			100	μA
I_{CC}	Power supply current			0.8	4	mA
	Standby				500	μA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. For X1/CLK, this swing is between 0.4V and 4.4V. All time measurements are referenced at input voltages of 0.8V and 2V and output voltages of 0.8V and 2V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. Test condition for outputs: $C_L = 150pF$, except interrupt outputs. Test conditions for interrupt outputs: $C_L = 50pF$, $R_L = 2.7k\Omega$ to V_{CC}.
8. Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the 'strobing' input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN. CEN and RDN (also CEN and WRN) are OR'ed internally. As a consequence, the signal asserted last initiates the cycle and the signal negated first terminates the cycle.
9. If CEN is used as the 'strobing' input, this parameter defines the minimum high time between one CEN and the next. The RDN signal must be negated for t_{RWD} to guarantee that any status register changes are valid.
10. Consecutive write operations to the same command require at least three edges of the X1 clock between writes.

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$[4, 5, 6, 7]

SYMBOL	PARAMETER	TENTATIVE LIMITS			UNIT
		Min	Typ	Max	
Reset timing (Figure 1)					
t_{RES}	RESET pulse width	100			ns
Bus timing (Figure 2)[8]					
t_{AS}	A0 – A2 setup time to RDN, WRN low	10			ns
t_{AH}	A0 – A2 hold time from RDN, WRN high	0			ns
t_{CS}	CEN setup time to RDN, WRN low	0			ns
t_{CH}	CEN hold time from RDN, WRN high	0			ns
t_{RW}	WRN, RDN pulse width	225			ns
t_{DD}	Data valid after RDN low			180	ns
t_{DF}	Data bus floating after RDN high			65	ns
t_{DS}	Data setup time before WRN high	100			ns
t_{DH}	Data hold time after WRN high	30			ns
t_{RWD}[10]	Time between READs and/or WRITEs	200			ns
MPI and MPO timing (Figure 3)[8]					
t_{PS}	MPI input setup time before RDN low	0			ns
t_{PH}	MPI input hold time after RDN high	0			ns
t_{PD}	MPO output valid after WRN high			370	ns
Interrupt timing (Figure 4)					
t_{IR}	INTRN negated:				
	Read RHR (RxRDY/FFULL interrupt)			370	ns
	Write THR (TxRDY, TxEMT interrupt)			370	ns
	Reset command (Break change interrupt)			370	ns
	Reset command (MPI change interrupt)			370	ns
	Stop C/T command (counter interrupt)			370	ns
	Write IMR (clear of interrupt mask bit)			270	ns
Clock timing (Figure 5)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2.0	3.6864	4.0	MHz
t_{CTC}	Counter/timer clock high or low time	100			ns
f_{CTC}	Counter/timer clock frequency	100		4.0M	Hz
t_{RX}	RxC high or low time	220			ns
f_{RX}	RxC frequency (16×)	100		2.0M	Hz
	(1×)	100		1.0M	Hz
t_{TX}	TxC high or low time	220			ns
f_{TX}	TxC frequency (16×)	0		2.0M	Hz
	(1×)	0		1.0M	Hz
Transmitter timing (Figure 6)					
t_{TXD}	TxD output delay from TxC low			350	ns
t_{TCS}	TxC output delay from TxD output data	0		150	ns
Receiver timing (Figure 7)					
t_{RXS}	RxD data setup time to RxC high	240			ns
t_{RXH}	RxD data hold time from RxC high	200			ns

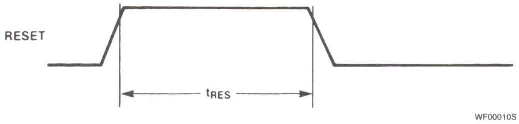

RESET
t_{RES}
WF00010S

Figure 1. Reset Timing

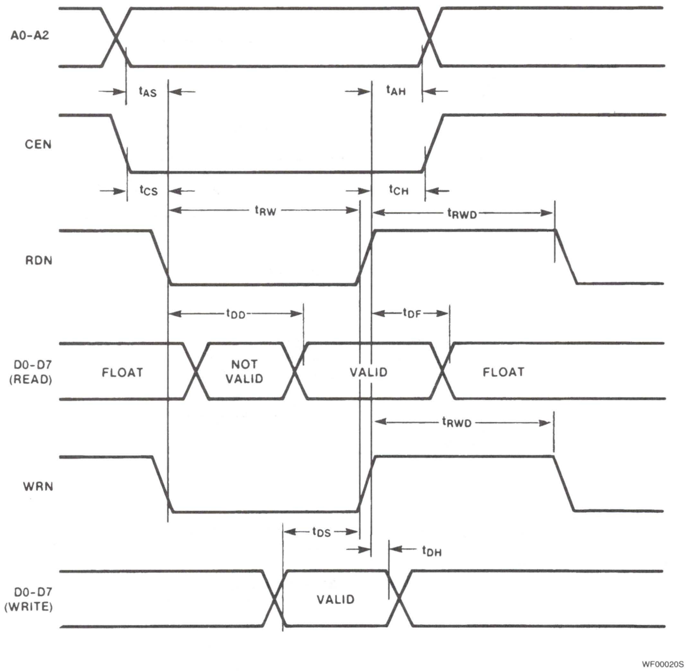

A0-A2
t_{AS}
t_{AH}
CEN
t_{CS}
t_{CH}
t_{RW}
t_{RWD}
RDN
t_{DD}
t_{DF}
D0-D7
(READ)
FLOAT
NOT VALID
VALID
FLOAT
t_{RWD}
WRN
t_{DS}
t_{DH}
D0-D7
(WRITE)
VALID
WF00020S

Figure 2. Bus Timing

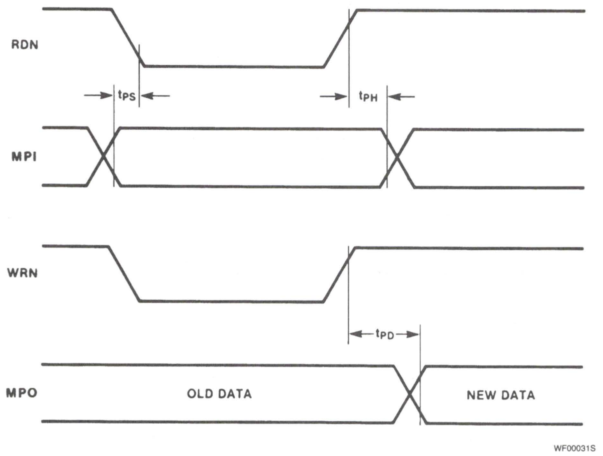
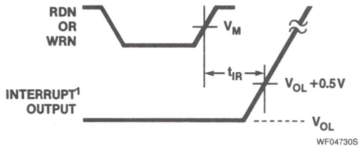

RDN
tPS
tPH
MPI
WRN
tPD
MPO
OLD DATA
NEW DATA
WF00031S

Figure 3. I/O Timing


RDN
OR
WRN
VM
tIR
INTERRUPT¹
OUTPUT
VOL +0.5V
VOL
WF04730S

NOTES:

1. INTRN or MPO when used as interrupt outputs.
2. The test for open drain outputs is intended to guarantee switching of the output transistor. Measurement of this response is referenced from the midpoint of the switching signal, V_M, to a point 0.5V above V_{OL}. This point represents a noise margin that assures true switching has occurred. Beyond this level, the effects of external circuitry and test environment are pronounced and can greatly affect the resultant measurement.

Figure 4. Interrupt Timing

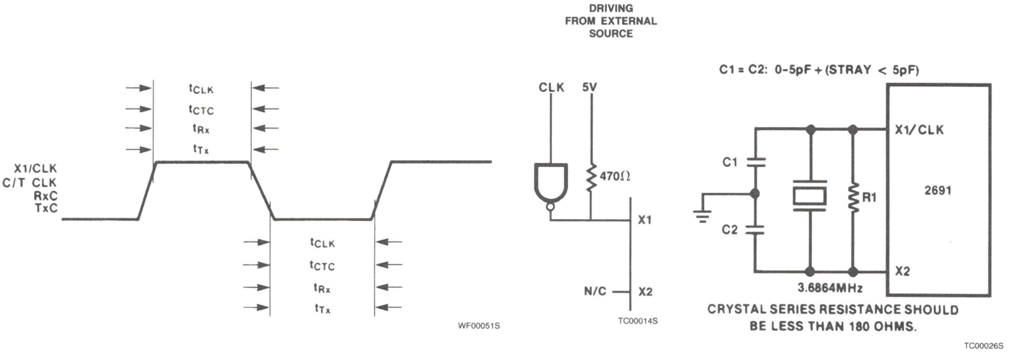

tCLK
tCTC
tRx
tTx
X1/CLK
C/T CLK
RxC
TxC
tCLK
tCTC
tRx
tTx
WF00051S
DRIVING
FROM EXTERNAL
SOURCE
CLK
5V
470Ω
X1
N/C
X2
TC00014S
C1 = C2: 0-5pF + (STRAY < 5pF)
X1/CLK
C1
2691
R1
C2
X2
3.6864MHz
CRYSTAL SERIES RESISTANCE SHOULD
BE LESS THAN 180 OHMS.
TC00026S

Figure 5. Clock Timing

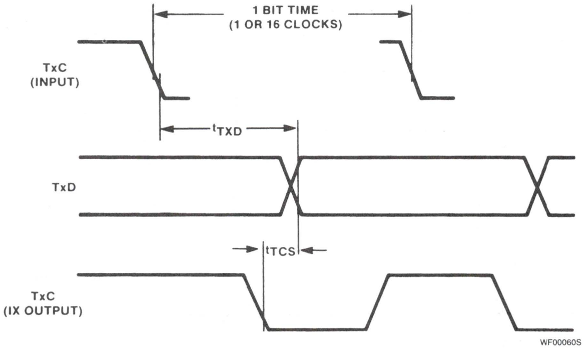

1 BIT TIME
(1 OR 16 CLOCKS)
TxC
(INPUT)
tTXD
TxD
tTCS
TxC
(IX OUTPUT)
WF00060S

Figure 6. Transmit Timing

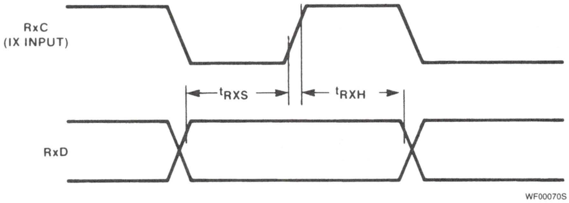
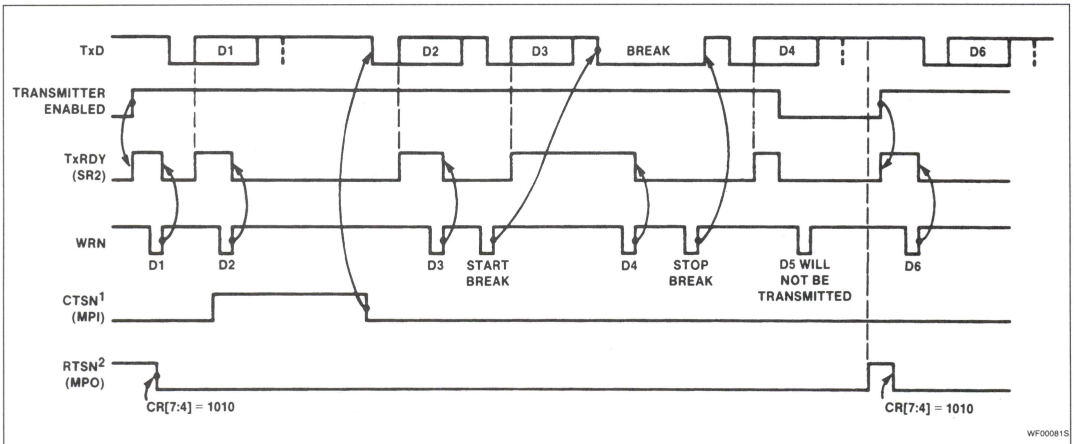
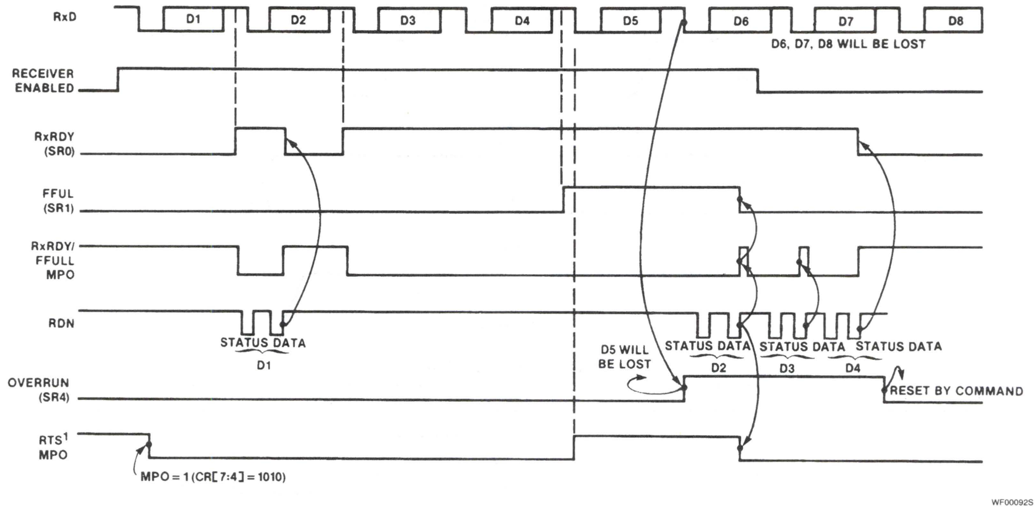

RxC
(IX INPUT)
tRXS
tRXH
RxD
WF00070S

Figure 7. Receive Timing


TxD
D1
D2
D3
BREAK
D4
D6
TRANSMITTER
ENABLED
TxRDY
(SR2)
WRN
D1
D2
D3
START
BREAK
D4
STOP
BREAK
D5 WILL
NOT BE
TRANSMITTED
D6
CTSN1
(MPI)
RTSN2
(MPO)
CR[7:4] = 1010
CR[7:4] = 1010
WF00081S

NOTES:
1. Timing shown for MR2[4] = 1.
2. Timing shown for MR2[5] = 1.

Figure 8. Transmitter Timing


RxD
D1
D2
D3
D4
D5
D6
D7
D8
D6, D7, D8 WILL BE LOST
RECEIVER ENABLED
RxRDY (SR0)
FFUL (SR1)
RxRDY/ FFULL MPO
RDN
STATUS DATA
D1
D5 WILL BE LOST
STATUS DATA
D2
STATUS DATA
D3
STATUS DATA
D4
OVERRUN (SR4)
RESET BY COMMAND
RTS1 MPO
MPO = 1 (CR[7:4] = 1010)
WF00092S

NOTES:
1. Timing shown for MR1[7] = 1.
2. Shown for ACR[2:0] = 111 and MR1[6] = 0.

Figure 9. Receiver Timing

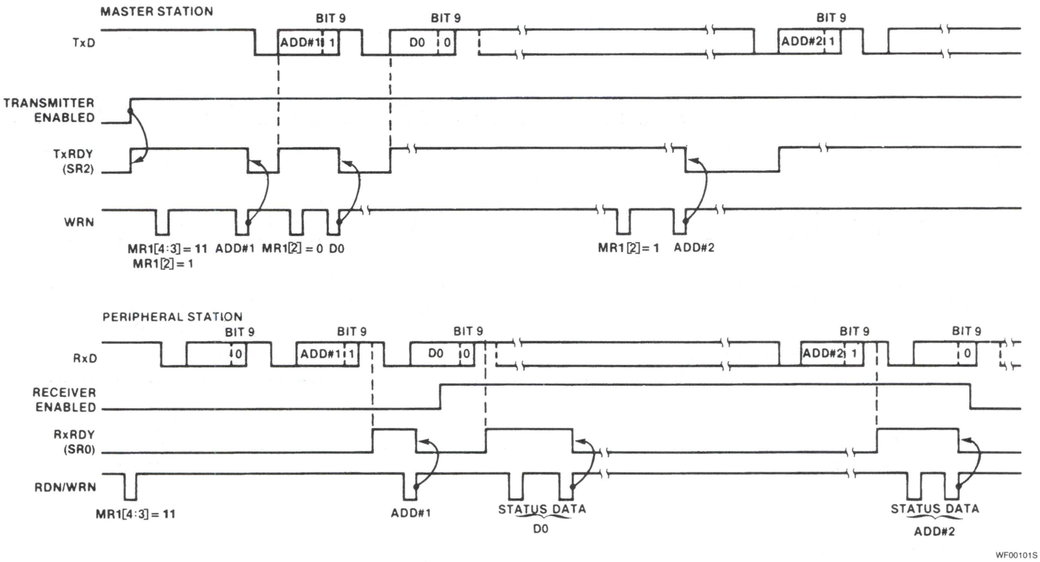

MASTER STATION
TxD
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
TRANSMITTER ENABLED
TxRDY (SR2)
WRN
MR1[4:3] = 11
MR1[2] = 1
ADD#1
MR1[2] = 0
D0
MR1[2] = 1
ADD#2
PERIPHERAL STATION
RxD
BIT 9
0
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
BIT 9
0
RECEIVER ENABLED
RxRDY (SR0)
RDN/WRN
MR1[4:3] = 11
ADD#1
STATUS DATA
D0
STATUS DATA
ADD#2
WF00101S

Figure 10. Wake-Up Mode

Signetics

Microprocessor Products

SCC2692
Dual Asynchronous Receiver/ Transmitter (DUART)

Preliminary Specification

DESCRIPTION

The Signetics SCC2692 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single-chip CMOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It interfaces directly with microprocessors and may be used in a polled or interrupt driven system.

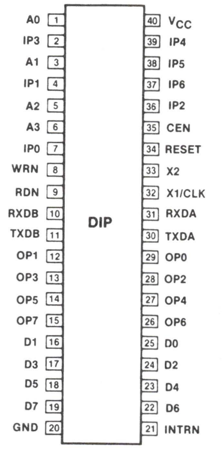
The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16× clock derived from a programmable counter/timer, or an external 1× or 16× clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

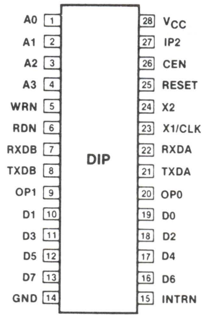
FEATURES

- **Dual full-duplex asynchronous receiver/transmitter**
- **Quadruple buffered receiver data registers**
- **Programmable data format**
  - **5 to 8 data bits plus parity**
  - **Odd, even, no parity or force parity**
  - **1, 1.5 or 2 stop bits programmable in 1/16-bit increments**
- **Programmable baud rate for each receiver and transmitter selectable from:**
  - **18 fixed rates: 50 to 38.4k baud**
  - **One user-defined rate derived from programmable timer/counter**
  - **External 1× or 16× clock**
- **Parity, framing, and overrun error detection**
- **False start bit detection**
- **Line break detection and generation**
- **Programmable channel mode**
  - **Normal (full-duplex)**
  - **Automatic echo**
  - **Local loopback**
  - **Remote loopback**
- **Multi-function programmable 16-bit counter/timer**
- **Multi-function 7-bit input port**
  - **Can serve as clock or control inputs**
  - **Change of state detection on four inputs**
- **Multi-function 8-bit output port**
  - **Individual bit set/reset capability**
  - **Outputs can be programmed to be status/interrupt signals**
- **Versatile interrupt system**
  - **Single interrupt output with eight maskable interrupting conditions**
  - **Output port can be configured to provide a total of up to six separate wire-ORable interrupt outputs**
- **Maximum data transfer: 1× - 1MB/s, 16× - 125kB/s**
- **Automatic wake-up mode for multidrop applications**
- **Start-end break interrupt/status**
- **Detects break which originates in the middle of a character**
- **On-chip crystal oscillator**
- **TTL compatible**
- **Single +5V power supply**
- **Power down mode**
- **Receiver timeout mode**

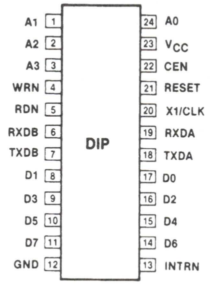
PIN CONFIGURATIONS


A0 1
IP3 2
A1 3
IP1 4
A2 5
A3 6
IP0 7
WRN 8
RDN 9
RXDB 10
TXDB 11
OP1 12
OP3 13
OP5 14
OP7 15
D1 16
D3 17
D5 18
D7 19
GND 20
DIP
40 VCC
39 IP4
38 IP5
37 IP6
36 IP2
35 CEN
34 RESET
33 X2
32 X1/CLK
31 RXDA
30 TXDA
29 OP0
28 OP2
27 OP4
26 OP6
25 D0
24 D2
23 D4
22 D6
21 INTRN

CD00130S

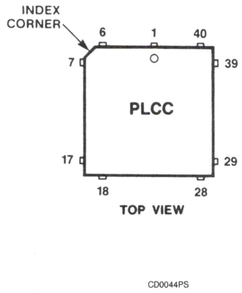

A0 1
A1 2
A2 3
A3 4
WRN 5
RDN 6
RXDB 7
TXDB 8
OP1 9
D1 10
D3 11
D5 12
D7 13
GND 14
DIP
28 VCC
27 IP2
26 CEN
25 RESET
24 X2
23 X1/CLK
22 RXDA
21 TXDA
20 OP0
19 D0
18 D2
17 D4
16 D6
15 INTRN

CD00140S


A1 1
A2 2
A3 3
WRN 4
RDN 5
RXDB 6
TXDB 7
D1 8
D3 9
D5 10
D7 11
GND 12
DIP
24 A0
23 VCC
22 CEN
21 RESET
20 X1/CLK
19 RXDA
18 TXDA
17 D0
16 D2
15 D4
14 D6
13 INTRN

CD00150S

PIN CONFIGURATIONS (Continued)



Pin	Function	Pin	Function
1	NC	23	NC
2	A0	24	INTRN
3	IP3	25	D6
4	A1	26	D4
5	IP1	27	D2
6	A2	28	D0
7	A3	29	OP6
8	IP0	30	OP4
9	WRN	31	OP2
10	RDN	32	OP0
11	RXDB	33	TXDA
12	NC	34	NC
13	TXDB	35	RXDA
14	OP1	36	X1/CLK
15	OP3	37	X2
16	OP5	38	RESET
17	OP7	39	CEN
18	D1	40	IP2
19	D3	41	IP6
20	D5	42	IP5
21	D7	43	IP4
22	GND	44	V_{CC}

ORDERING INFORMATION

PACKAGES	$V_{CC} = +5V \pm 5\%$, $T_A = 0°C$ to $+70°C$			
	24-Pin[1]	28-Pin[2]	40-Pin[2]	44-Pin
Ceramic DIP	Not available	SCC2692AC1I28	SCC2692AC1I40	Not available
Plastic DIP	SCC2692AC1N24	SCC2692AC1N28	SCC2692AC1N40	Not available
Plastic LCC	Not available	Not available	Not available	SCC2692AC1A44

NOTES:
1. 400 mil wide DIP
2. 600 mil wide DIP

Also provided on the SCC2692 are a multipurpose 7-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

The SCN2692 is available in four package versions: 40-pin and 28-pin, both 0.6″ wide DIPs; a compact 24-pin 0.4″ wide DIP; and a 44-pin PLCC.

PIN DESCRIPTION

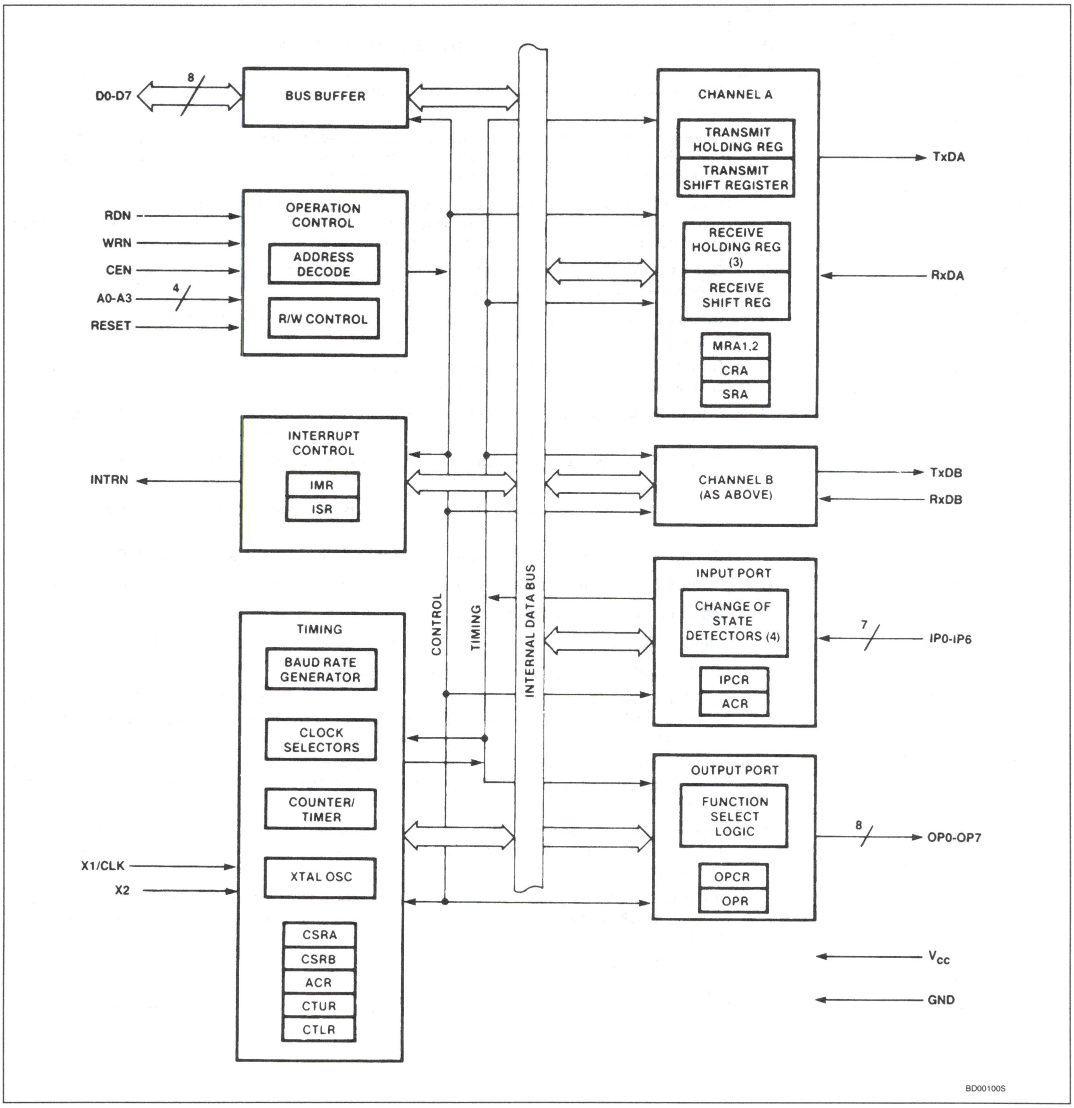
MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
D0 – D7	X	X	X	I/O	**Data Bus:** Bidirectional 3-State data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CEN	X	X	X	I	**Chip Enable:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0 – D7 as controlled by the WRN, RDN and A0 – A3 inputs. When high, places the D0 – D7 lines in the 3-State condition.
WRN	X	X	X	I	**Write Strobe:** When low and CEN is also low, the contents of the data bus is loaded into the addressed register. The transfer occurs on the rising edge of the signal.
RDN	X	X	X	I	**Read Strobe:** When low and CEN is also low, causes the contents of the addressed register to be presented on the data bus. The read cycle begins on the falling edge of RDN.
A0 – A3	X	X	X	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESET	X	X	X	I	**Reset:** A high level clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), puts OP0 – OP7 in the high state, stops the counter/timer, and puts channels A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
INTRN	X	X	X	O	**Interrupt Request:** Active low, open drain, output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
X1/CLK	X	X	X	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. When a crystal is used, a capacitor must be connected from this pin to ground (see Figure 5).
X2	X	X		I	**Crystal 2:** Connection for other side of the crystal. When a crystal is used, a capacitor must be connected from this pin to ground (see Figure 5). If X1/CLK is driven from an external source, this pin must be open.
RxDA	X	X	X	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.

PIN DESCRIPTION (Continued)

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
RxDB	X	X	X	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	X	X	X	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	X	X	X	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	X	X		O	**Output 0:** General purpose output, or channel A request to send (RTSAN, active low). Can be deactivated automatically on receive or transmit.
OP1	X	X		O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low). Can be deactivated automatically on receive or transmit.
OP2	X			O	**Output 2:** General purpose output, or channel A transmitter 1× or 16× clock output, or channel A receiver 1× clock output.
OP3	X			O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1× clock output, or channel B receiver 1× clock output.
OP4	X			O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.
OP5	X			O	**Output 5:** General purpose output, or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	X			O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output.
OP7	X			O	**Output 7:** General purpose output, or channel B open drain, active low, TxRDYB output.
IP0	X			I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN).
IP1	X			I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN).
IP2	X	X		I	**Input 2:** General purpose input, or counter/timer external clock input.
IP3	X			I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP4	X			I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	X			I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP6	X			I	**Input 6:** General purpose input or channel B receiver external clock input (RxCB). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
V_{CC}	X	X	X	I	**Power Supply:** +5V supply input
GND	X	X	X	I	**Ground**

Dual Asynchronous Receiver/Transmitter (DUART) SCC2692

BLOCK DIAGRAM


D0-D7
8
BUS BUFFER
RDN
WRN
CEN
A0-A3
4
RESET
OPERATION CONTROL
ADDRESS DECODE
R/W CONTROL
INTRN
INTERRUPT CONTROL
IMR
ISR
TIMING
BAUD RATE GENERATOR
CLOCK SELECTORS
COUNTER/ TIMER
XTAL OSC
CSRA
CSRB
ACR
CTUR
CTLR
X1/CLK
X2
CONTROL
TIMING
INTERNAL DATA BUS
CHANNEL A
TRANSMIT HOLDING REG
TRANSMIT SHIFT REGISTER
RECEIVE HOLDING REG (3)
RECEIVE SHIFT REG
MRA1,2
CRA
SRA
TxDA
RxDA
CHANNEL B (AS ABOVE)
TxDB
RxDB
INPUT PORT
CHANGE OF STATE DETECTORS (4)
IPCR
ACR
7
IP0-IP6
OUTPUT PORT
FUNCTION SELECT LOGIC
OPCR
OPR
8
OP0-OP7
V_{CC}
GND
BD00100S

BLOCK DIAGRAM

The 2692 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurrence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions.

Outputs OP3 – OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART.

If an external clock is used instead of a crystal, X1 should be driven using a configuration similar to the one in Figure 5. The input clock must be capable of attaining a V_{IH} of 4.4V.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4k baud. The clock outputs from the BRG are at 16× the actual baud rate. The counter/timer can be used as a timer to produce a 16× clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection, for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Communications Channels A And B

Each communications channel of the 2692 comprises a full-duplex asynchronous receiver/transmitter (UART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The input port pulse detection circuitry uses a 38.4kHz sampling clock derived from one of the baud rate generator taps. This results in a sampling period of slightly more than 25μs (this assumes that the clock input is 3.6864MHz). The detection circuitry, in order to guarantee that a true change in level has occurred, requires two successive samples at the new logic level be observed. As a consequence, the minimum duration of the signal change is 25μs if the transition occurs "coincident with the first sample pulse." The 50μs time refers to the situation in which the change of state is "just missed" and the first change of state is not detected until 25μs later.

The inputs to this unlatched 7-bit port can be read by the CPU by performing a read operation at address D16. A high input results in a logic 1 while a low input results in a logic 0. D7 will always be read as a logic 1. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs, lasting longer than 25 – 50μs, will set the corresponding bit in the input port change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E16 with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F16 with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The 2692 is conditioned to transmit data when the transmitter is enabled through the command register. The 2692 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted. If it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted, if the transmitter has been disabled.

Receiver

The 2692 is conditioned to receive data when enabled through the command register. The receiver looks for a high-to-low (mark-to-space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16× clock for 7½ clocks (16× clock mode) or at the next rising edge of the bit time clock (1× clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are "popped", thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the "character" mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the "block" mode, the status provided in the SR for these three bits is the logical-OR of the status for all characters coming to the top of the FIFO since the last "reset error" command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is "popped" only when the RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected; the character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set-upon receipt of the start bit of the new (overrunning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Timeout Mode

Under certain conditions, the user may want to set the receiver to interrupt the CPU when the receive FIFO becomes full. This can be accomplished by programming MR1[6] = 1. If a message that is only one or two characters long is received, the FIFO is not full so that ISR[1] does not set and the CPU is not interrupted. The CPU will not know that there is data in the receive FIFO. The timeout mode provides the user with a timeout interrupt via the C/T. If a character is received and the FIFO does not become full, a pre-selected period of delay can be timed out by the C/T and the CPU interrupted.

This mode is enabled by writing the appropriate command to the command register. Writing an "AX" to CRA or CRB will invoke the timeout mode for that channel. Writing a "CX" to CRA or CRB will reset the timeout mode. CTU and CTL must be loaded with a value greater than the normal receive character period. Each time a received character is transferred from the shift register to the RHR, the C/T is reloaded with the value in CTU and CTL and then restarted. If the C/T is allowed to end the count, the counter ready bit (ISR[3]) will be set. If IMR[3] is set, an interrupt will occur.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to '11' for channels A and B, respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, with receivers that are normally disabled, examine the received data stream and 'wakeup' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

Table 1. 2692 Register Addressing

A3	A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	Rx Holding Register A (RHRA)	Tx Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	Rx Holding Register B (RHRB)	Tx Holding Register B (THRB)
1	1	0	0	*Reserved*	*Reserved*
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in Table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems.

For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x, switches the pointer to MR2x. The pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to Table 2 for register bit descriptions. The reserved registers at addresses H'02' and H'0A' should never be read during normal operation since they are reserved for internal diagnostics.

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control

This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select

This bit selects either the channel A receiver ready status (RxRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the accumulation (logical-OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select

If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select

This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits Per Character Select

This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

Table 2. Register Bit Formats

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RxRTS CONTROL	RxINT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHARACTER	
MR1A **MR1B**	0 = no 1 = yes	0 = RxRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = multi-drop mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

Table 2. Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	CHANNEL MODE		TxRTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A **MR2B**	00 = Normal 01 = Auto-echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

NOTE:
*Add 0.5 to values shown for 0–7 if channel is programmed for 5 bits/char.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
CSRA **CSRB**	See Text				See Text			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	MISCELLANEOUS COMMANDS				DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
CRA **CRB**	See Text				0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA **SRB**	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

NOTE:
*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits (4:0). These bits are cleared by a 'reset error status' command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1 = TxRDYB	0 = OPR[6] 1 = TxRDYA	0 = OPR[5] 1 = RxRDY/ FFULLB	0 = OPR[4] 1 = RxRDY/ FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB(1×) 11 = RxCB(1×)		00 = OPR[2] 01 = TxCA(16×) 10 = TxCA(1×) 11 = RxCA(1×)	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See Table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	DELTA IP3	DELTA IP2	DELTA IP1	DELTA IP0	IP3	IP2	IP1	IP0
IPCR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	INPUT PORT CHANGE	DELTA BREAK B	RxRDY/ FFULLB	TxRDYB	COUNTER READY	DELTA BREAK A	RxRDY/ FFULLA	TxRDYA
ISR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

Table 2. Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IMR	**IN. PORT CHANGE INT**	**DELTA BREAK B INT**	**RxRDY/ FFULLB INT**	**TxRDYB INT**	**COUNTER READY INT**	**DELTA BREAK A INT**	**RxRDY/ FFULLA INT**	**TxRDYA INT**
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTUR	**C/T[15]**	**C/T[14]**	**C/T[13]**	**C/T[12]**	**C/T[11]**	**C/T[10]**	**C/T[9]**	**C/T[8]**

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTLR	**C/T[7]**	**C/T[6]**	**C/T[5]**	**C/T[4]**	**C/T[3]**	**C/T[2]**	**C/T[1]**	**C/T[0]**

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select

Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission; i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is relocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control

This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR 2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Verify the message is sent by waiting until the transmit ready status (TxRDY) is asserted. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-send Control
If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select
This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of $\frac{9}{16}$ to 1 and 1$\frac{9}{16}$ to 2 bits, in increments of $\frac{1}{16}$ bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1$\frac{1}{16}$ to 2 stop bits can be programmed in increments of $\frac{1}{16}$-bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1× clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select
This field selects the baud rate clock for the channel A receiver as follows:

	Baud Rate	
CSRA[7:4]	**ACR[7] = 0**	**ACR[7] = 1**
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4k	19.2k
1 1 0 1	Timer	Timer
1 1 1 0	IP4 — 16×	IP4 — 16×
1 1 1 1	IP4 — 1×	IP4 — 1×

The receiver clock is always a 16× clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select
This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRA[3:0]	**ACR[7] = 0**	**ACR[7] = 1**
1 1 1 0	IP3 — 16×	IP3 — 16×
0 1 1 1	IP3 — 1×	IP3 — 1×

The transmitter clock is always a 16× clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select
This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[7:4]	**ACR[7] = 0**	**ACR[7] = 1**
1 1 1 0	IP6 — 16×	IP6 — 16×
0 1 1 1	IP6 — 1×	IP6 — 1×

The receiver clock is always a 16× clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select
This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[3:4]	**ACR[7] = 0**	**ACR[7] = 1**
1 1 1 0	IP5 — 16×	IP5 — 16×
1 1 1 1	IP5 — 1×	IP5 — 1×

The transmitter clock is always a 16× clock except for CSRB[3:0] = 1111.

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands
The encoded value of this field may be used to specify a single command as follows:

CRA[7:4]	COMMAND
0 0 0 0	No command.
0 0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.
0 0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
0 1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
0 1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
0 1 1 0	Start break. Forces the TxDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.

CRA[7:4]	COMMAND
0 1 1 1	Stop Break. The TxDA line will go high (marking) within two bit times. TxDA will remain high for one bit time before the next character, if any, is transmitted.
1 0 0 0	Assert RTSN. Causes the RTSN output to be asserted (low).
1 0 0 1	Negate RTSN. Causes the RTSN output to be negated (high).
1 0 1 0	Set special timeout mode on with this channel as the channel to restart the C/T as each receive character is transferred from the serial shift register to the RHR.
1 0 1 1	Not used.
1 1 0 0	Reset special timeout mode.
1 1 0 1	Not used.
1 1 1 0	Power Down mode on. In this mode, the DUART oscillator is stopped and all functions requiring this clock are suspended. The contents of all registers are saved. It is recommended that the transmitter and receiver be disabled prior to placing the DUART is this mode. This bit is reset with the assertion of RESET. This command is in CRA only.
1 1 1 1	Power Down Mode Normal Run. This command resets the Power Down mode. This command is in CRA only.

CRA[3] — Disable Channel A Transmitter
This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter
Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver
This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver
Enables operation of the channel A receiver. If not in the special wakeup mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA with the exceptions of the last two command "EX" and "FX" which are used for power down mode. These two commands are not used in CRB. All other control actions that apply to CRA also apply to CRB.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break
This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received: further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1× clock).

When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error
This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error
This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error
This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set-upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA)
This bit will be set when the channel A transmitter underruns; i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA)
This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

SRA[1] — Channel A FIFO Full (FFULLA)
This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA)
This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, only if (after this read) there are no more characters in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select
This bit programs the OP7 output to provide one of the following:

–The complement of OPR[7]

–The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select
This bit programs the OP6 output to provide one of the following:

- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select
This bit programs the OP5 output to provide one of the following:

- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select
This bit programs the OP4 output to provide one of the following:

- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select
This field programs the OP3 output to provide one of the following:

- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open-drain output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1× clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1× clock is output.
- The 1× clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1× clock is output.

OPCR[1:0] — OP2 Output Select
This field programs the OP2 output to provide one of the following:

- The complement of OPR[2]
- The 16× clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be a 1× clock if CSRA[3:0] = 1111.
- The 1× clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1× clock is output.
- The 1× clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1× clock is output.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select
This bit selects one of two sets of baud rates to be generated by the BRG:

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05k, 1.2k, 2.4k, 4.8k, 7.2k, 9.6k, and 38.4k baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k, and 19.2k baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in Table 3.

ACR[6:4] — Counter/Timer Mode And Clock Source Select
This field selects the operating mode of the counter/timer and its clock source as shown in Table 4.

ACR[3:0] — IP3, IP2, IP1, IP0 Change Of State Interrupt Enable
This field selects which bits of the input port change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the "on" state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the "off" state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change-of-State
These bits are set when a change-of-state, as defined in the input port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State
These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initialized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status
This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

ISR[6] — Channel B Change In Break
This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B "reset break change interrupt" command.

ISR[5] — Channel B Receiver Ready Or FIFO Full
The function of this bit is programmed by MR1B[6]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is "popped". If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full; i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[4] — Channel B Transmitter Ready
This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready
In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter

Table 3. Baud Rate Generator Characteristics Crystal or Clock = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16× CLOCK (kHz)	ERROR (%)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2k	307.2	0
38.4k	614.4	0

NOTE:
Duty cycle of 16× clock is 50% ±1%

Table 4. ACR 6:4 Field Definition

ACR 6:4	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)
0 0 1	Counter	TxCA — 1× clock of channel A transmitter
0 1 0	Counter	TxCB — 1× clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)
1 0 1	Timer	External (IP2) divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change In Break
This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready Or FIFO Full
The function of this bit is programmed by MR1A[6]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready
This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3 - OP7 or the reading of the ISR.

CTUR And CTLR — Counter/ Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs, respectively, of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 0002_{16}. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A3 - A0 = 1110) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR.

The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A3 - A0 = 1111). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

On power up and after reset, the timer/ counter runs in timer mode and can only be restarted. Because it cannot be shut off or stopped, and runs continuously in timer mode, it is recommended that at initialization, the output port (OP3) should be masked off through the OPCR[3:2] = 00 until the T/C is programmed to the desired operational state.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8 bits to the upper 8 bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	0 to +70	°C
T_{STG}	Storage temperature	-65 to +150	°C
V_{CC}	Voltage from V_{CC} to GND[3]	−0.5 to +7	V
V_S	Voltage from any pin to ground[3]	−0.5 to V_{CC} ±10%	V
P_D	Power dissipation	21	mW

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5.0V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage					
	All except X1/CLK		2			V
	X1/CLK		$0.9V_{CC}$		V_{CC}	V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}[7]	Output high voltage					
	(except open drain outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	µA
I_{LL}	Data bus 3-State leakage current	$V_O = 0.4$ to V_{CC}	−10		10	µA
I_{OD}	Open drain output leakage current	$V_O = 0.4$ to V_{CC}	−10		10	µA
I_{X1L}	X1/CLK low input current	$V_{IN} = 0$, X2 floated	−100	-3.0	0.0	µA
I_{X1H}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 floated	0.0	+30	100	µA
When oscillator is in power-down mode:						
I_{X1H}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 floated	2	6	10	mA
I_{CC}	Power supply current			0.8	4	mA
	Standby				500	µA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. For X1/CLK this swing is between 0.4V and 4.4V. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. Test condition for outputs: $C_L = 150pF$, except interrupt outputs. Test conditions for interrupt outputs: $C_L = 50pF$, $R_L = 2.7k\Omega$ to V_{CC}.
8. Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operrated with CEN as the 'strobing' input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN. CEN and RDN (also CEN and WRN) are OR'ed internally. As a consequence, the signal asserted last initiates the cycle and the signal negated first terminates the cycle.
9. If CEN is used as the 'strobing' input, this parameter defines the minimum high time between one CEN and the next. The RDN signal must be negated for t_{RWD} to guarantee that any status register changes are valid.
10. Consecutive write operations to the same command require at least three edges of the X1 clock between writes.

Dual Asynchronous Receiver/Transmitter (DUART) SCC2692

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5.0V \pm 5\%$[4, 5, 6, 7]

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Typ	Max	
Reset timing (Figure 1)					
t_{RES}	RESET pulse width	1.0			µs
Bus timing (Figure 2)[8]					
t_{AS}	A0 – A3 setup time to RDN, WRN low	10			ns
t_{AH}	A0 – A3 hold time from RDN, WRN high	0			ns
t_{CS}	CEN setup time to RDN, WRN low	0			ns
t_{CH}	CEN hold time from RDN, WRN high	0			ns
t_{RW}	WRN, RDN pulse width	225			ns
t_{DD}	Data valid after RDN low			180	ns
t_{DF}	Data bus floating after RDN high			65	ns
t_{DS}	Data setup time before WRN high	100			ns
t_{DH}	Data hold time after WRN high	0			ns
t_{RWD}	High time between READs and/or WRITEs[9,10]	200			ns
Port timing (Figure 3)[8]					
t_{PS}	Port input setup time before RDN low	0			ns
t_{PH}	Port input hold time after RDN high	0			ns
t_{PD}	Port output valid after WRN high			370	ns
Interrupt timing (Figure 4)					
t_{IR}	INTRN (or OP3 – OP7 when used as interrupts) negated from:				
	Read RHR (RxRDY/FFULL interrupt)			370	ns
	Write THR (TxRDY interrupt)			370	ns
	Reset command (delta break interrupt)			370	ns
	Stop C/T command (counter interrupt)			370	ns
	Read IPCR (input port change interrupt)			370	ns
	Write IMR (clear of interrupt mask bit)			370	ns
Clock timing (Figure 5)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2.0	3.6864	4	MHz
t_{CTC}	CTCLK (IP2) high or low time	100			ns
f_{CTC}	CTCLK (IP2) frequency	100		4M	Hz
t_{RX}	RxC high or low time	220			ns
f_{RX}	RxC frequency (16×)	100		2M	Hz
	(1×)	100		1M	Hz
t_{TX}	TxC high or low time	220			ns
f_{TX}	TxC frequency (16×)	0		2	MHz
	(1×)	0		1	MHz
Transmitter timing (Figure 6)					
t_{TXD}	TxD output delay from TxC low			350	ns
t_{TCS}	Output delay from TxC low to TxD data output	0		150	ns
Receiver timing (Figure 7)					
t_{RXS}	RxD data setup time to RxC high	240			ns
t_{RXH}	RxD data hold time from RxC high	200			ns

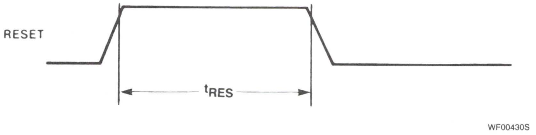

RESET
tRES
WF00430S

Figure 1. Reset Timing

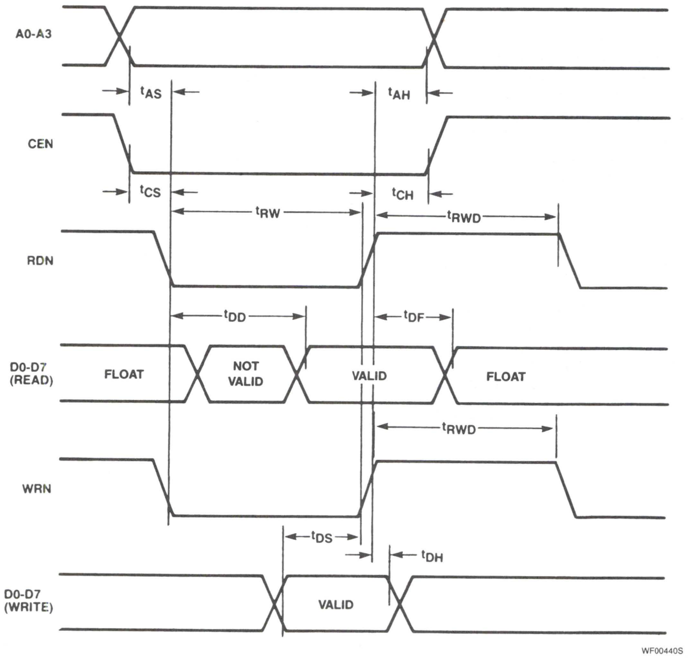

A0-A3
tAS
tAH
CEN
tCS
tCH
tRW
tRWD
RDN
tDD
tDF
D0-D7
(READ)
FLOAT
NOT
VALID
VALID
FLOAT
tRWD
WRN
tDS
tDH
D0-D7
(WRITE)
VALID
WF00440S

Figure 2. Bus Timing

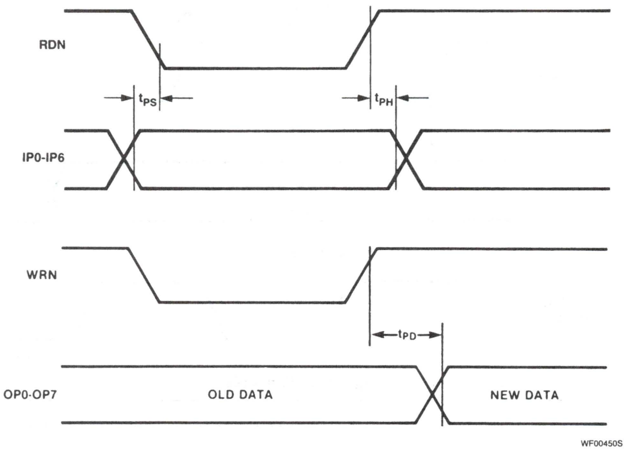
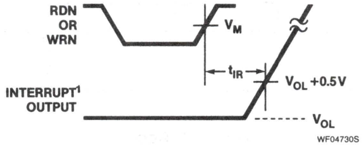

RDN
tPS
tPH
IP0-IP6
WRN
tPD
OP0-OP7
OLD DATA
NEW DATA
WF00450S

Figure 3. Port Timing


RDN
OR
WRN
VM
tIR
INTERRUPT1
OUTPUT
VOL +0.5V
VOL
WF04730S

NOTES:

1. INTRN or OP3 – OP7 when used as interrupt outputs.
2. The test for open drain outputs is intended to guarantee switching of the output transistor. Measurement of this response is referenced from the midpoint of the switching signal, V_M, to a point 0.5V above V_{OL}. This point represents noise margin that assures true switching has occurred. Beyond this level, the effects of external circuitry and test environment are pronounced and can greatly affect the resultant measurement.

Figure 4. Interrupt Timing

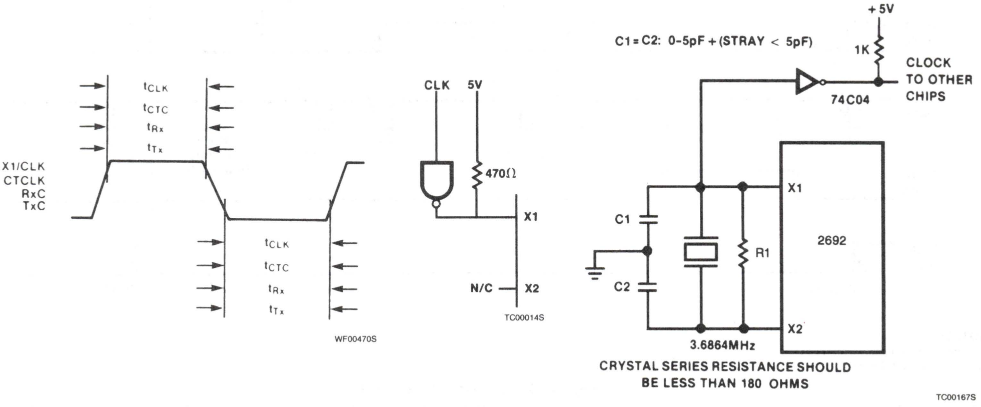

tCLK
tCTC
tRx
tTx
X1/CLK
CTCLK
RxC
TxC
tCLK
tCTC
tRx
tTx
WF00470S
CLK
5V
470Ω
X1
N/C
X2
TC00014S
+5V
C1 = C2: 0-5pF + (STRAY < 5pF)
1K
CLOCK
TO OTHER
CHIPS
74C04
X1
C1
R1
2692
C2
X2
3.6864MHz
CRYSTAL SERIES RESISTANCE SHOULD
BE LESS THAN 180 OHMS
TC00167S

Figure 5. Clock Timing

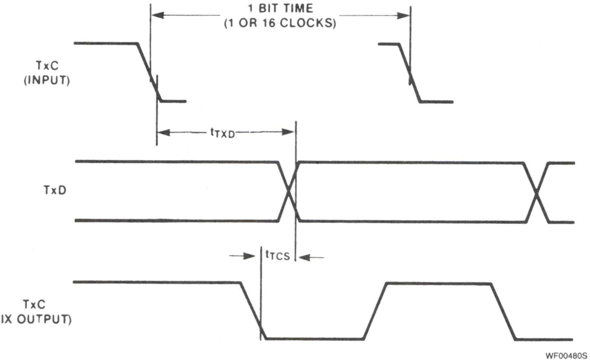

1 BIT TIME
(1 OR 16 CLOCKS)
TxC
(INPUT)
tTXD
TxD
tTCS
TxC
(IX OUTPUT)
WF00480S

Figure 6. Transmit

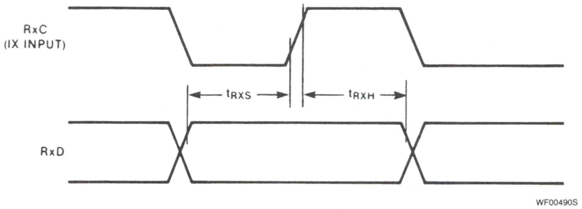
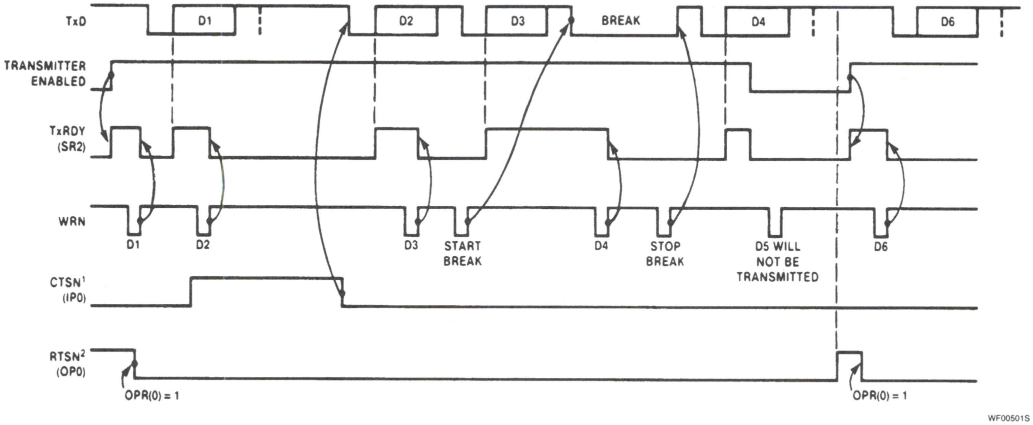
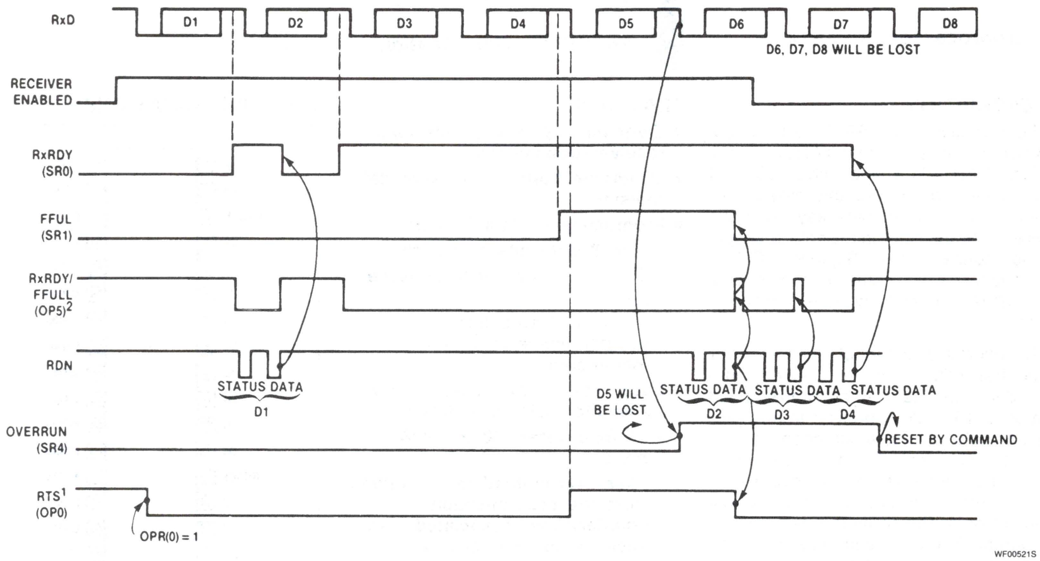

RxC
(IX INPUT)
tRXS
tRXH
RxD
WF00490S

Figure 7. Receive


TxD
D1
D2
D3
BREAK
D4
D6
TRANSMITTER
ENABLED
TxRDY
(SR2)
WRN
D1
D2
D3
START
BREAK
D4
STOP
BREAK
D5 WILL
NOT BE
TRANSMITTED
D6
CTSN¹
(IP0)
RTSN²
(OP0)
OPR(0) = 1
OPR(0) = 1
WF00501S

NOTES:
1. Timing shown for MR2(4) = 1.
2. Timing shown for MR2(5) = 1.

Figure 8. Transmitter Timing


RxD
D1
D2
D3
D4
D5
D6
D7
D8
D6, D7, D8 WILL BE LOST
RECEIVER ENABLED
RxRDY (SR0)
FFUL (SR1)
RxRDY/ FFULL (OP5)2
RDN
STATUS DATA
D1
D5 WILL BE LOST
STATUS DATA STATUS DATA STATUS DATA
D2 D3 D4
OVERRUN (SR4)
RESET BY COMMAND
RTS1 (OP0)
OPR(0) = 1
WF00521S

NOTES:
1. Timing shown for MR1(7) = 1.
2. Shown for OPCR(4) = 1 and MR(6) = 0.

Figure 9. Receiver Timing

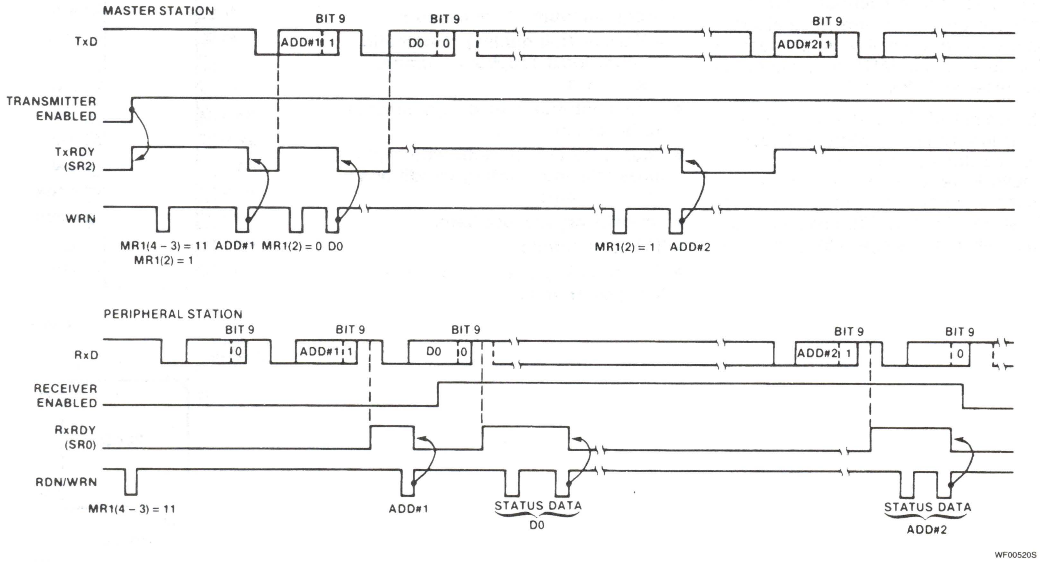

MASTER STATION
TxD
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
TRANSMITTER ENABLED
TxRDY (SR2)
WRN
MR1(4 – 3) = 11 ADD#1 MR1(2) = 0 D0
MR1(2) = 1
MR1(2) = 1 ADD#2
PERIPHERAL STATION
RxD
BIT 9
0
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
BIT 9
0
RECEIVER ENABLED
RxRDY (SR0)
RDN/WRN
MR1(4 – 3) = 11
ADD#1
STATUS DATA
D0
STATUS DATA
ADD#2
WF00520S

Figure 10. Wake-Up Mode

Signetics

Microprocessor Products

SCC2698
Octal Universal Asynchronous Receiver/Transmitter (Octal-UART)

Preliminary Specification

DESCRIPTION

The Signetics SCC2698 Octal Universal Asynchronous Receiver/Transmitter (Octal-UART) is a single-chip MOS-LSI communications device that provides an eight-channel full-duplex asynchronous receiver/transmitter in a single package. It is fabricated with Signetics' CMOS technology, which combines the benefits of high density and low power consumption.

The operating speed of each receiver and transmitter can be selected independently as one of eighteen fixed baud rates, a 16× clock derived from a programmable counter/timer, or an external 1× or 16× clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the Octal-UART particularly functional for dual-speed channel applications such as clustered terminal systems.

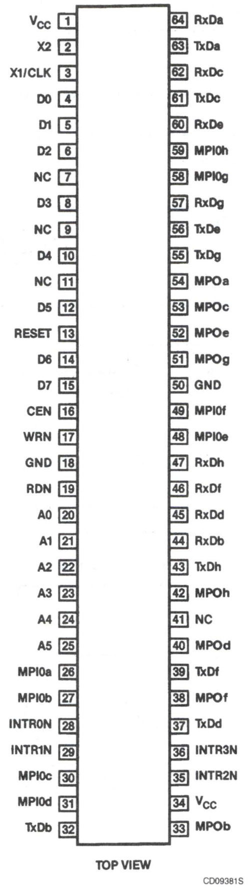
The receiver is quadruply buffered to minimize the potential of receiver overrun, or to reduce interrupt overhead in interrupt driven systems. In addition, a handshaking capability is provided to disable a remote UART transmitter when the receiver buffer is full.

The Octal-UART provides a power-down mode in which the oscillator is frozen but the register contents are stored. This results in reduced power consumption on the order of several magnitudes. The Octal-UART is fully TTL compatible and operates from a single +5V power supply.

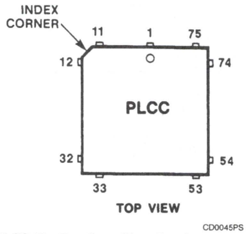
FEATURES

- **Eight full-duplex asynchronous receiver/transmitters**
- **Quadruple buffered receiver data register**
- **Programmable data format:**
  - **5 to 8 data bits plus parity**
  - **Odd, even, no parity or force parity**
  - **1, 1.5 or 2 stop bits programmable in 1/16-bit increments**
- **Baud rate for the receiver and transmitter selectable from:**
  - **18 fixed rates: 50 to 38.4k baud**
  - **Four user-defined rates derived from the programmable counter/timer associated with each of the four blocks**
  - **External 1× or 16× clock**
- **Parity, framing, and overrun error detection**
- **False start bit detection**
- **Line break detection and generation**
- **Programmable channel mode**
  - **Normal (full-duplex), automatic echo, local loopback, remote loopback**
- **Four multi-function programmable 16-bit counter/timers**
- **Four interrupt outputs with eight maskable interrupting conditions for each output**
- **On-chip crystal oscillator**
- **TTL compatible**
- **Single +5V power supply with low power mode**

PIN CONFIGURATIONS



V_{CC} 1
X2 2
X1/CLK 3
D0 4
D1 5
D2 6
NC 7
D3 8
NC 9
D4 10
NC 11
D5 12
RESET 13
D6 14
D7 15
CEN 16
WRN 17
GND 18
RDN 19
A0 20
A1 21
A2 22
A3 23
A4 24
A5 25
MPI0a 26
MPI0b 27
INTR0N 28
INTR1N 29
MPI0c 30
MPI0d 31
TxDb 32
64 RxDa
63 TxDa
62 RxDc
61 TxDc
60 RxDe
59 MPI0h
58 MPI0g
57 RxDg
56 TxDe
55 TxDg
54 MPOa
53 MPOc
52 MPOe
51 MPOg
50 GND
49 MPI0f
48 MPI0e
47 RxDh
46 RxDf
45 RxDd
44 RxDb
43 TxDh
42 MPOh
41 NC
40 MPOd
39 TxDf
38 MPOf
37 TxDd
36 INTR3N
35 INTR2N
34 V_{CC}
33 MPOb
TOP VIEW
CD09381S



INDEX CORNER
11
1
75
12
74
PLCC
32
54
33
53
TOP VIEW
CD0045PS

PLCC Pin Functions (Next Page)

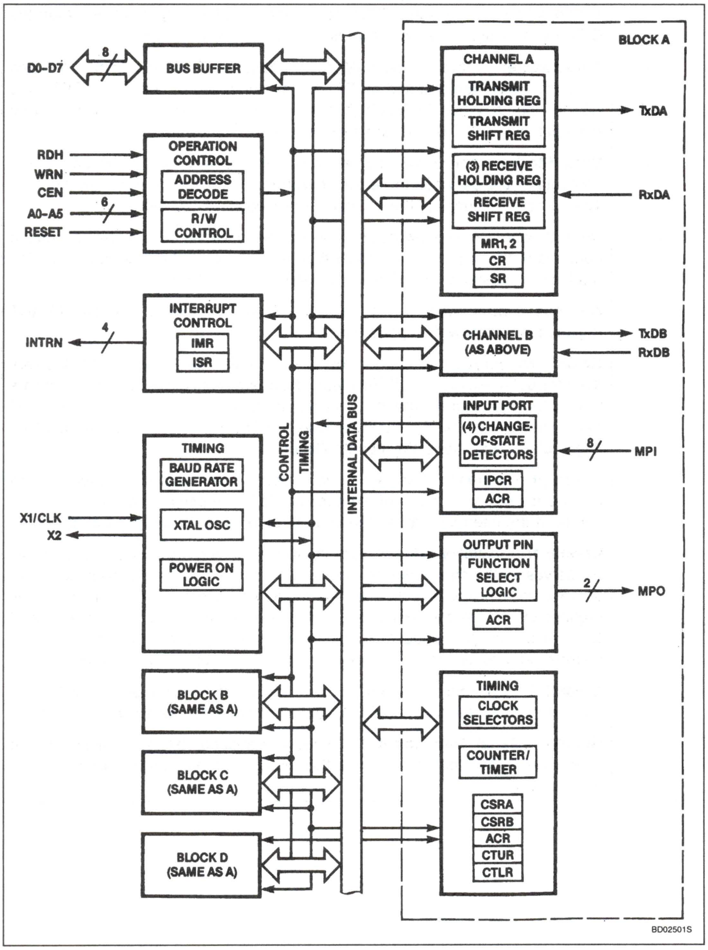
Octal Universal Asynchronous Receiver/Transmitter (Octal-UART)

SCC2698

ORDERING INFORMATION

PACKAGES	$V_{CC} = +5V \pm 5\%$, $T_A = 0$ to $+70°C$
Plastic DIP	SCC2698AC1N64
Plastic LCC	SCC2698AC1A84

BLOCK DIAGRAM


D0-D7
8
BUS BUFFER
RDH
WRN
CEN
A0-A5
6
RESET
OPERATION CONTROL
ADDRESS DECODE
R/W CONTROL
INTRN
4
INTERRUPT CONTROL
IMR
ISR
TIMING
BAUD RATE GENERATOR
X1/CLK
X2
XTAL OSC
POWER ON LOGIC
BLOCK B (SAME AS A)
BLOCK C (SAME AS A)
BLOCK D (SAME AS A)
CONTROL
TIMING
INTERNAL DATA BUS
BLOCK A
CHANNEL A
TRANSMIT HOLDING REG
TRANSMIT SHIFT REG
TxDA
(3) RECEIVE HOLDING REG
RECEIVE SHIFT REG
RxDA
MR1, 2
CR
SR
CHANNEL B (AS ABOVE)
TxDB
RxDB
INPUT PORT
(4) CHANGE-OF-STATE DETECTORS
IPCR
ACR
8
MPI
OUTPUT PIN FUNCTION SELECT LOGIC
ACR
2
MPO
TIMING
CLOCK SELECTORS
COUNTER/TIMER
CSRA
CSRB
ACR
CTUR
CTLR
BD02501S

PIN CONFIGURATIONS (Continued)

PLCC

Pin	Function	Pin	Function
1	TxDa	43	MPOb
2	MPI13g	44	MPI2d
3	RxDa	45	V_{CC}
4	MPI13h	46	INTR2N
5	V_{CC}	47	INTR3N
6	X2	48	MPI3c
7	X1/CLK	49	TxDd
8	D0	50	MPI3d
9	D1	51	MPOd
10	D2	52	TxDf
11	D3	53	MPOf
12	D4	54	MPOh
13	D5	55	TxDh
14	MPI1a	56	RxDb
15	RESET	57	RxDd
16	D6	58	RxDf
17	D7	59	RxDh
18	CEN	60	MPI1e
19	WRN	61	MPI0e
20	GND	62	MPI1f
21	MPI1b	63	MPI0f
22	RDN	64	MPI2e
23	A0	65	GND
24	MPI2a	66	MPI2f
25	A1	67	MPOg
26	MPI2b	68	MPI3e
27	A2	69	MPOe
28	MPI3a	70	MPI3f
29	A3	71	MPOc
30	MPI3b	72	MPOa
31	A4	73	TxDg
32	A5	74	TxDe
33	MPI0a	75	RxDg
34	MPI0b	76	MPI0g
35	INTR0N	77	MPI0h
36	INTR1N	78	MPI1g
37	MPI0c	79	RxDe
38	MPI1c	80	MPI1h
39	MPI0d	81	TxDc
40	MPI1d	82	MPI2g
41	TxDb	83	RxDc
42	MPI2c	84	MPI2h

Octal Universal Asynchronous Receiver/Transmitter (Octal-UART)

SCC2698

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
D0 – D7	4 – 6, 8, 10, 12, 14, 15	8 – 13, 16, 17	I/O	**Data Bus:** Active high 8-bit bidirectional 3-State data bus. Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the Octal-UART take place over this bus. The direction of the transfer is controlled by the WRN and RDN inputs when the CEN input is low. When the CEN input is high, the data bus is in the 3-State condition.
CEN	16	18	I	**Chip Enable:** Active low input. When low, data transfers between the CPU and the Octal-UART are enabled on D0 – D7 as controlled by the WRN, RDN, and A0 – A5 inputs. When CEN is high, the Octal-UART is effectively isolated from the data bus and D0 – D7 are placed in the 3-State condition.
WRN	17	19	I	**Write Strobe:** Active low input. A low on this pin while CEN is low causes the contents of the data bus to be transferred to the register selected by A0 – A5. The transfer occurs on the trailing (rising) edge of the signal.
RDN	19	22	I	**Read Strobe:** Active low input. A low on this pin while CEN is low causes the contents of the register selected by A0 – A5 to be placed on the data bus. The read cycle begins on the leading (falling) edge of RDN.
A0 – A5	20 – 25	23, 25, 27, 29, 31, 32	I	**Address Inputs:** Active high address inputs to select the Octal-UART registers for read/write operations.
RESET	13	15	I	**Reset:** Master reset. A high on this pin clears the status register (SR), clears the interrupt mask register (IMR), clears the interrupt status register (ISR), clears the output port configuration register (OPCR), places the receiver and transmitter in the inactive state causing the TxD output to go to the marking (high) state, and stops the counter/timer.
INTR0N – INTR3N	28, 29, 35, 36	35, 36, 46, 47	O	**Interrupt Request:** This active low open drain output is asserted on occurrence of one or more of eight maskable interrupting conditions. The CPU can read the interrupt status register to determine the interrupting condition(s).
X1/CLK	3	7	I	**Crystal 1:** Crystal or external clock input. When using the crystal oscillator, this pin serves as the connection for one side of the crystal. If a crystal is not used, an external clock is supplied at this input. An external clock (or crystal) is required even if the internal baud rate generator is not utilized. This clock is used to drive the internal baud rate generator, as an optional input to the timer/counter, and to provide other clocking signals required by the chip.
X2	2	6	I	**Crystal 2:** Connection for other side of crystal. If an external source is used instead of a crystal, this should be open or connected as shown in Figure 6.
RxDa – RxDh	64, 44, 62, 45, 60, 46, 57, 47	3, 56, 83, 57, 79, 58, 75, 59		**Receiver Serial Data Input:** The least significant bit is received first. If external receiver clock is specified, this input is sampled on the rising edge of the clock.

Octal Universal Asynchronous Receiver/Transmitter (Octal-UART)

SCC2698

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
TxDa – TxDh	63, 32, 61, 37, 56, 39, 55, 43	1, 41, 81, 49, 74, 52, 73, 55	O	**Transmitter Serial Data Output:** Transmitter serial data output. The least significant bit is transmitted first. This output is held in the marking (high) condition when the transmitter is idle or disabled and when the Octal-UART is operating in local loopback mode. If external transmitter clock is specified, the data is shifted on the falling edge of the transmitter clock.
MPOa – MPOh	54, 33, 53, 38, 52, 40, 51, 42	72, 43, 71, 51, 69, 53, 67, 54	O	**Multi-Purpose Output:** One of the following functions can be selected for this output pin by programming the output port configuration register: **RTSN** — Request to send active low output. This output is asserted and negated via the command register. By appropriate programming of the mode registers, RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted or when the receiver FIFO and shift register are full. **C/TO** — Counter/timer output **TxC1×** — 1× clock for the transmitter **TxC16×** — 16× clock for the transmitter **RxC1×** — 1× clock for the receiver **RxC16×** — 16× clock for the receiver **TxRDY** — Transmitter holding register empty signal **RxRDY/FFULL** — Receiver FIFO not empty/full signal
MPI0a – MPI0h	26, 27, 30, 31, 48, 49, 58, 59	33, 34, 37, 39, 61, 63, 76, 77	I	**Multi-Purpose Input 0:** This pin can be programmed to serve as an input for one of the following functions: **GPI** — General purpose input. The current state of the pin can be determined by reading the IPCR or input port register. **CTSN** — Clear-to-Send active low input.
MPI1a – MPI1h	NC	14, 21, 38, 40, 60, 62, 78, 80	I	**Multi-Purpose Input 1:** This pin can be programmed to serve as an input for one of the following functions: **GPI** — General purpose input. The current state of the pin can be determined by reading the IPCR or input port register. **CTCLK** — Counter/timer external clock input. Only channels a, c, e, and g change to C/T inputs; channels b, d, f and h stays GPI.
MPI2a – MPI2h	NC	24, 26, 42, 44, 64, 66, 82, 84	I	**Multi-Purpose Input 2:** This pin can be programmed to serve as an input for one of the following functions: **GPI** — General purpose input. The current state of the pin can be determined by reading the input port register. **TCLK** — Transmitter external clock input. This may be a 1× or 16× clock as programmed by CSR[3:0].
MPI3a – MPI3h	NC	28, 30, 48, 50, 68, 70, 2, 4	I	**Multi-Purpose Input 3:** This pin can be programmed to serve as an input for one of the following functions: **GPI** — General purpose input. The current state of the pin can be determined by reading the input port register. **RCLK** — Receiver external clock input. This may be a 1× or 16× clock as programmed by CSR[7:4].
V_{CC}	1, 34	5, 45	I	**Power Supply:** +5V supply input
GND	18, 50	20, 65	I	**Ground**

BLOCK DIAGRAM

As shown on the block diagram, the Octal-UART consists of: data bus buffer, interrupt control, operation control, timing, and eight receiver and transmitter channels. The eight channels are divided into four different blocks each block independent of each other (see Figure 1).

BLOCK A CHANNELS a, b	BLOCK C CHANNELS e, f
BLOCK B CHANNELS c, d	BLOCK D CHANNELS g, h

Figure 1. Channel Architecture

Channel Blocks

There are four blocks (Figure 1), each containing two sets of receiver/transmitters. In the following discussion, the description applies to block A which contains channels a and b. However, the same information applies to all channel blocks.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data buses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the Octal-UART.

Interrupt Control

A single interrupt output (INTRN) is provided which is asserted on the occurrence of any of the following internal events:

- Transmit holding register ready for each channel
- Receive holding register ready or FIFO full for each channel
- Change in break received status for each channel
- Counter reached terminal count
- Change in MPI input

Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR can be programmed to select only certain conditions, of the above, to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. However, the bits of the ISR are not masked by the IMR.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are shown in Table 1.

Mode registers 1 and 2 are accessed via an auxiliary pointer. The pointer is set to MR1 by RESET or by issuing a reset pointer command via the command register. Any read or write of the mode register while the pointer is at MR1 switches the pointer to MR2. The pointer then remains at MR2 so that subsequent accesses are to MR2, unless the pointer is reset to MR1 as described above.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer for each block, and two clock selectors.

The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs with a minimum of external components. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. If an external clock is used instead of a crystal, both X1 and X2 are driven using a configuration similar to the one in Figure 6. Also, an arrangement where X2 is floating can be used, however, the input high voltage must be capable of attaining 4.4V. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock frequency, within the limits specified in the electrical specifications, must be supplied even if the internal BRG is not used.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4k baud. Thirteen of these are available simultaneously for use by the receiver and transmitter. Eight are fixed, and one of two sets of five can be selected by programming ACR[7]. The clock outputs from the BRG are at 16× the actual baud rate. The counter/timer can be used as a timer to produce a 16× clock for any other baud rate by counting down the crystal clock or an external clock. The clock selectors allow the independent selection, by the receiver and transmitter, of any of these baud rates or an external timing signal.

There are four C/Ts in the Octal-UART, one for each block. The C/T operation is programmed by ACR[6:4]. One of eight timing sources can be used as the input to the C/T. The output of the C/T is available to the clock selectors and can also be programmed by OPCR[2:0] for channel a and OPCR[6:4] for channel b, to be output on the MPOa or MPOb pin, respectively.

In the timer mode, the C/T generates a square wave whose period is twice the number of clock periods loaded into the C/T upper and lower registers. The counter ready bit in the ISR is set once each cycle of the square wave. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be affected. In this mode, the C/T runs continuously and does not recognize the stop C/T command (the command only resets the counter ready bit in the ISR). Receipt of a start C/T command causes the C/T to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR. Counting begins upon receipt of a start counter command. Upon reaching terminal count, the counter ready bit in the ISR is set. The counter continues counting past the terminal count until stopped by the CPU. If MPO is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and the counter ready bit is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command following a stop counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower eight bits of the counter may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems, which may occur, if a carry from the lower eight bits to the upper eight bits occurs between the times both halves of the counter are read. However, a subsequent start counter command causes the counter to begin a new count cycle using the values in CTUR and CTLR.

Receiver and Transmitter

The Octal-UART has eight full-duplex asynchronous receiver/transmitters. The operating frequency for the receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

Registers associated with the communications channel are the mode registers (MR1 and MR2), the clock select register (CSR), the command register (CR), the status register (SR), the transmit holding register (THR), and the receive holding register (RHR).

Transmitter

The transmitter accepts parallel data from the CPU and converts it to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is

Table 1. Register Addressing

A5	A4	A3	A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	0	0	0	MR1a, MR2a	MR1a, MR2a
0	0	0	0	0	1	SRa	CSRa
0	0	0	0	1	0	Reserved*	CRa
0	0	0	0	1	1	RHRa	THRa
0	0	0	1	0	0	IPCRA	ACRA
0	0	0	1	0	1	ISRA	IMRA
0	0	0	1	1	0	CTUA	CTURA
0	0	0	1	1	1	CTLA	CTLRA
0	0	1	0	0	0	MR1b, MR2b	MR1b, MR2b
0	0	1	0	0	1	SRb	CSRb
0	0	1	0	1	0	Reserved*	CRb
0	0	1	0	1	1	RHRb	THRb
0	0	1	1	0	0	Reserved*	Reserved*
0	0	1	1	0	1	Input port A	OPCRA
0	0	1	1	1	0	Start C/T A	Reserved*
0	0	1	1	1	1	Stop C/T A	Reserved*
0	1	0	0	0	0	MR1c, MR2c	MR1c, MR2c
0	1	0	0	0	1	SRc	CSRc
0	1	0	0	1	0	Reserved*	CRc
0	1	0	0	1	1	RHRc	THRc
0	1	0	1	0	0	IPCRB	ACRB
0	1	0	1	0	1	ISRB	IMRB
0	1	0	1	1	0	CTUB	CTURB
0	1	0	1	1	1	CTLB	CTLRB
0	1	1	0	0	0	MR1d, MR2d	MR1d, MR2d
0	1	1	0	0	1	SRd	CSRd
0	1	1	0	1	0	Reserved*	CRd
0	1	1	0	1	1	RHRd	THRd
0	1	1	1	0	0	Reserved*	Reserved*
0	1	1	1	0	1	Input port B	OPCRB
0	1	1	1	1	0	Start C/T B	Reserved*
0	1	1	1	1	1	Stop C/T B	Reserved*
1	0	0	0	0	0	MR1e, MR2e	MR1e, MR2e
1	0	0	0	0	1	SRe	CSRe
1	0	0	0	1	0	Reserved*	CRe
1	0	0	0	1	1	RHRe	THRe
1	0	0	1	0	0	IPCRC	ACRC
1	0	0	1	0	1	ISRC	IMRC
1	0	0	1	1	0	CTUC	CTURC
1	0	0	1	1	1	CTLC	CTLRC
1	0	1	0	0	0	MR1f, MR2f	MR1f, MR2f
1	0	1	0	0	1	SRf	CSRf
1	0	1	0	1	0	Reserved*	CRf
1	0	1	0	1	1	RHRf	THRf
1	0	1	1	0	0	Reserved*	Reserved*
1	0	1	1	0	1	Input port C	OPCRC
1	0	1	1	1	0	Start C/T C	Reserved*
1	0	1	1	1	1	Stop C/T C	Reserved*
1	1	0	0	0	0	MR1g, MR2g	MR1g, MR2g
1	1	0	0	0	1	SRg	CSRg
1	1	0	0	1	0	Reserved*	CRg
1	1	0	0	1	1	RHRg	THRg
1	1	0	1	0	0	IPCRD	ACRD
1	1	0	1	0	1	ISRD	IMRD
1	1	0	1	1	0	CTUD	CTURD
1	1	0	1	1	1	CTLD	CTLRD
1	1	1	0	0	0	MR1h, MR2h	MR1h, MR2h
1	1	1	0	0	1	SRh	CSRh
1	1	1	0	1	0	Reserved*	CRh
1	1	1	0	1	1	RHRh	THRh
1	1	1	1	0	0	Reserved*	Reserved*
1	1	1	1	0	1	Input port D	OPCRD
1	1	1	1	1	0	Start C/T D	Reserved*
1	1	1	1	1	1	Stop C/T D	Reserved*

NOTES:

*Reserved registers should never be read during normal operation since they are reserved for internal diagnostics.

ACR = Auxiliary control register
CR = Command register
CSR = Clock select register
CTL = Counter/timer lower
CTLR = Counter/timer lower register
CTU = Counter/timer upper
CTUR = Counter/timer upper register
MR = Mode register
SR = Status register
THR = Tx holding register
RHR = Rx holding register
IPCR = Input port change register
ISR = Interrupt status register
IMR = Interrupt mask register
OPCR = Output port configuration register

not available in the THR, the TxD output remains high and the TxEMT bit in the SR will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. In the 16× clock mode, this also resynchronizes the internal 1× transmitter clock so that transmission of the new character begins with minimum delay.

The transmitter can be forced to send a break (continuous low condition) by issuing a start break command via the CR. The break is terminated by a stop break command. If the transmitter is disabled, it continues operating until the character currently being transmitted and the character in the THR, if any, are completely sent out. Characters cannot be loaded into the THR while the transmitter is disabled.

Receiver

The receiver accepts serial data on the RxD pin, converts the serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition, and presents the assembled character to the CPU. The receiver looks for a high-to-low (mark-to-space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled again each 16× clock for 7½ clocks (16× clock mode) or at the next rising edge of the bit time clock (1× clock mode).

If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver samples the input. This continues at one-bit time intervals, at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the RHR and the RxRDY bit in the SR is set to a one. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (i.e. framing error) and RxD remains low for one-half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled). The parity error, framing error and overrun error (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set.

If a break condition is detected (RxD is low for the entire character including the stop bit), only one character consisting of all zeros will be loaded into the FIFO and the received break bit in the SR is set to 1. The RxD input must return to a high condition for two successive clock edges of the 1× clock (internal or external) before a search for the next start bit begins.

TIME OUT MODE

Under certain conditions the user may want to set the receiver to interrupt the CPU when the receiver FIFO becomes full. This can be accomplished by programming MR1[6] = 1. If a message that is only one or two characters long is received, the FIFO is not full so that ISR[1] does not set and the CPU is not interrupted. The CPU will not know that there is data in the receive FIFO. The time-out mode provides the user with a time-out interrupt via the C/T. If a character is received and the FIFO does not become full, a pre-selected period of delay can be timed out by the C/T and the CPU interrupted.

This mode is enabled by writing the appropriate command to the command register. Writing an "AX" to CRA or CRB will invoke the time-out mode for that channel. Writing a "CX" to CRA or CRB will reset the time-out mode. CTU and CTL must be loaded with a value greater than the normal receive character period. Each time a received character is transferred from the shift register to the RHR, the C/T is reloaded with the value in CTU and CTL and then restarted. If the C/T is allowed to end the count, the counter ready bit (ISR[3]) will be set. If IMR[3] is set, an interrupt will occur.

RECEIVER FIFO

The RHR consists of a first-in-first-out (FIFO) with a capacity of three characters. Data is loaded from the receive shift register into the top-most empty position of the FIFO. The RxRDY bit, in the status register (SR), is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR, outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits are 'popped', thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are appended to each data character in the FIFO. Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical-OR of the status for all characters coming to the top of the FIFO, since the last reset error command was issued. In either mode, reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore, the SR should be read prior to reading the corresponding data character.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exists, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit, SR[4], will be set on receipt of the start bit of the new (overrunning) character.

WAKE-UP MODE

In addition to the normal transmitter and receiver operation described above, the Octal-UART incorporates a special mode which provides automatic wake-up of the receiver through address frame recognition for multiprocessor communications. This mode is selected by programming bits MR1[4:3] to '11'.

In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, whose receivers are normally disabled, examine the received data stream and 'wake up' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1[2]; MR1[2] = 0 transmits a zero in the A/D bit position which identifies the corresponding data bits as data, MR1[2] = 1 transmits a one in the A/D bit position which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

While in this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one, but discards the received character if the received A/D bit is a zero. If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SR[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

MULTI-PURPOSE INPUT PIN

The inputs to this unlatched 8-bit port for each block can be read by the CPU, by performing a read operation as shown in Table 1. A high input results in a logic one while a low input results in a logic zero. When the input port pins are read on the 84-pin PLCC, they will appear on the data bus in alternating pairs. (i.e., DB0 = MPI0a, DB1 = MPI1a, DB2 = MPI0b, DB3 = MPI1b, DB4 = MPI2a, DB5 = MPI3a, DB6 = MPI2b, DB7 = MPI3b. Although this example is shown for input port 'A', all input ports will have a similar order).

The MPI pins can be programmed as an input to one of several Octal-UART circuits. The function of the pin is selected by programming the appropriate control register. Change-of-state detectors are provided for MPI0 and MPI1 for each channel in each block. A high-to-low or low-to-high transition of the inputs lasting longer than 25 to 50μs sets the MPI change-of-state bit in the interrupt status register. The bit is cleared via a command. The change-of-state can be programmed to generate an interrupt to the CPU, by setting the corresponding bit in the interrupt mask register.

The input port pulse detection circuitry uses a 38.4kHz sampling clock, derived from one of the baud rate generator taps. This produces a sampling period of slightly more than 25μs (assuming a 3.6864MHz oscillator input). The detection circuitry, in order to guarantee that a true change in level has occurred, requires two successive samples be observed at the new logic level. As a consequence, the minimum duration of the signal change is 25μs if the transition occurs coincident with the first sample pulse. (The 50μs time refers to the condition where the change-of-state is just missed and the first change-of-state is not detected until after an additional 25μs.)

MULTI-PURPOSE OUTPUT PIN

This pin can be programmed to serve as a request-to-send output, the counter/timer output, the output for the 1× or 16× transmitter or receiver clocks, the TxRDY output or the RxRDY/FFULL output (see OPCR [2:0] and OPCR [6:4] — MPO Output Select).

REGISTERS

The operation of the Octal-UART is programmed by writing control words into the appropriate registers. Operational feedback is provided via registers which can be read by the CPU. Addressing the registers is described in Table 1.

If the contents of the MR, the CSR, the OPCR and ACR are changed while the receiver(s) and transmitter(s) are enabled, the registers will not be updated until both the transmitter(s) are empty and the receiver(s) disabled. The receiver(s) will be disabled, at the completion of the character being received, at the time of the register change. Normally these registers should not be updated without first making sure that the receiver(s) and transmitter(s) are disabled, and the C/T is stopped.

The bit formats of the Octal-UART registers are depicted in Table 2. These are shown for block A. The bit format for the other blocks is the same.

MR1 — Mode Register 1

MR1 is accessed when the MR pointer points to MR1. The pointer is set to MR1 by RESET or by a set pointer command applied via CR. After reading or writing MR1, the pointers are set at MR2.

MR1[7] — Receiver Request-to-Send Control

This bit controls the deactivation of the RTSN output (MPO) by the receiver. This output is manually asserted and negated by commands applied via the command register. MR1[7] = 1 causes RTSN to be automatically negated upon receipt of a valid start bit if the receiver FIFO is full. RTSN is reasserted when an empty FIFO position is available. This feature can be used to prevent overrun in the receiver, by using the RTSN output signal, to control the CTS input of the transmitting device.

MR1[6] — Receiver Interrupt Select

This bit selects either the receiver ready status (RxRDY) or the FIFO full status (FFULL) to be used for CPU interrupts.

MR1[5] — Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break). In the character mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the block mode, the status provided in the SR for these bits is the accumulation (logical-OR) of the status for all characters coming to the top of the FIFO, since the last reset error command was issued.

MR1[4:3] — Parity Mode Select

If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1[4:3] = 11 selects the channel to operate in the special wake-up mode.

MR1[2] — Parity Type Select

This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special 'wake-up' mode, it selects the polarity of the A/D bit.

MR1[1:0] — Bits per Character Select

This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2 — Mode Register 2

MR2 is accessed when the channel a MR pointer points to MR2, which occurs after any access to MR1. Accesses to MR2 do not change the pointer.

MR2[7:6] — Mode Select

The Octal-UART can operate in one of four modes: MR2[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU-to-receiver communications continue normally, but the CPU-to-transmitter link is disabled.

Two diagnostic modes can also be selected. MR2[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU-to-transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., the transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

When switching in and out of the various modes, the selected mode is activated at the completion of all transmitted and received characters. Likewise, if a mode is deselected, the device will switch out of the mode at the completion of the current transmit and/or receive characters.

MR2[5] — Transmitter Request-to-Send Control

This bit controls the deactivation of the RTSN output (MPO) by the transmitter. This output is manually asserted and negated by appropriate commands issued via the command register. MR2[5] = 1 causes RTSN to be reset automatically one bit time after the characters in the transmit shift register and in the THR (if any) are completely transmitted (includes the programmed number of stop bits if the transmitter is not enabled). This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2[5] = 1.
2. Enable transmitter.
3. Assert RTSN via command.
4. Send message.
5. Verify the next to last character of the message is being sent by waiting until transmitter ready is asserted. Disable transmitter after the last character is loaded into the THR.
6. The last character will be transmitted and RTSN will be reset one bit time after the last stop bit.

MR2[4] — Clear-to-Send Control

The state of this bit determines if the CTSN input (MPI) controls the operation of the transmitter. If this bit is 0, CTSN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSN each time it is ready to send a character. If it is asserted (low), the character is transmitted. If it is negated (high), the TxD output remains in the marking state and the transmission is delayed until CTSN goes low. Changes in CTSN, while a character is being transmitted, do not affect the transmission of that character. This feature can be used to prevent overrun of a remote receiver.

MR2[3:0] — Stop Bit Length Select

This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of $^9/_{16}$ to 1 and $1^9/_{16}$ to 2 bits, in increments of $^1/_{16}$ bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, $1^1/_{16}$ to 2 stop bits can be programmed in increments of $^1/_{16}$ bit. In all cases, the receiver only checks for a mark condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled). If an external 1× clock is used for the transmitter, MR2[3] = 0 selects one stop bit and MR2[3] = 1 selects two stop bits to be transmitted.

Table 2. Register Bit Formats

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RxRTS CONTROL	**RxINT SELECT**	**ERROR MODE**	**PARITY MODE**		**PARITY TYPE**	**BITS PER CHARACTER**	
MR1	0 = no 1 = yes	0 = RxRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = special mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	CHANNEL MODE		**TxRTS CONTROL**	**CTS ENABLE Tx**	**STOP BIT LENGTH***			
MR2	00 = Normal 01 = Auto-echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

NOTE:
*Add 0.5 to values shown for 0 – 7, if channel is programmed for 5 bits/character.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVER CLOCK SELECT				**TRANSMITTER CLOCK SELECT**			
CSR	See Text				See Text			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	MISCELLANEOUS COMMANDS				**DISABLE Tx**	**ENABLE Tx**	**DISABLE Rx**	**ENABLE Rx**
CR	See Text				0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

Table 2. Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SR	**RECEIVED BREAK**	**FRAMING ERROR**	**PARITY ERROR**	**OVERRUN ERROR**	**TxEMT**	**TxRDY**	**FFULL**	**RxRDY**
	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

NOTE:
*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits [7:5] from the top of the FIFO together with bits [4:0]. These bits are cleared by a reset error status command. In character mode, they must be reset when the corresponding data character is read from the FIFO.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OPCR		**MPOb PIN FUNCTION SELECT**			**POWER-DOWN MODE**	**MPOa PIN FUNCTION SELECT**		
	not used	000 = RTSN 001 = C/TO 010 = TxC (1×) 011 = TxC (16×) 100 = RxC (1×) 101 = RxC (16×) 110 = TxRDY 111 = RxRDY/FF			0 = off 1 = on	000 = RTSN 001 = C/TO 010 = TxC (1×) 011 = TxC (16×) 100 = RxC (1×) 101 = RxC (16×) 110 = TxRDY 111 = RxRDY/FF		

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ACR	**BRG SET SELECT**	**COUNTER/TIMER MODE AND SOURCE**			**DELTA MPI1bINT**	**DELTA MPI0bINT**	**DELTA MPI1aINT**	**DELTA MPI0aINT**
	0 = set 1 1 = set 2	See Text			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IPCR	**DELTA MPI1b**	**DELTA MPI0b**	**DELTA MPI1a**	**DELTA MPI0a**	**MPI1b**	**MPI0b**	**MPI1a**	**MPI0a**
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ISR	**MPI PORT CHANGE**	**DELTA BREAKb**	**RxRDY/ FFULLb**	**TxRDYb**	**COUNTER READY**	**DELTA BREAKa**	**RxRDY/ FFULLa**	**TxRDYa**
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IMR	**MPI PORT CHANGE INT**	**DELTA BREAKb INT**	**RxRDY/ FFULLb INT**	**TxRDYb INT**	**COUNTER READY INT**	**DELTA BREAKa INT**	**RxRDY/ FFULLa INT**	**TxRDYa INT**
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTUR	**C/T[15]**	**C/T[14]**	**C/T[13]**	**C/T[12]**	**C/T[11]**	**C/T[10]**	**C/T[9]**	**C/T[8]**

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTLR	**C/T[7]**	**C/T[6]**	**C/T[5]**	**C/T[4]**	**C/T[3]**	**C/T[2]**	**C/T[1]**	**C/T[0]**

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Input Port Register	**MPI3b**	**MPI2b**	**MPI3a**	**MPI2a**	**MPI1b**	**MPI0b**	**MPI1a**	**MPI0a**
	0 = low	0 = low	0 = low	0 = low	0 = low	0 = low	0 = low	0 = low
	1 = high	1 = high	1 = high	1 = high	1 = high	1 = high	1 = high	1 = high

CSR — Clock Select Register

CSR[7:4] — Receiver Clock Select
When using a 3.6864MHz crystal or external clock input, this field selects the baud rate clock for the receiver as shown in Table 3.

The receiver clock is always a 16× clock, except for CSR[7:4] = 1111. When MPI3 is selected as the input, MPI3a is for channel a and MPI3b is for channel b.

CSR[3:0] — Transmitter Clock Select
This field selects the baud rate clock for the transmitter. The field definition is as shown in Table 3 except as follows.

CSR[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	MPI2 — 16×	MPI2 — 16×
1 1 1 1	MPI2 — 1×	MPI2 — 1×

When MPI3 is selected as the input, MPI3a is for channel a and MPI3b is for channel b.

CR — Command Register

CR is used to write commands to the Octal-UART.

CR[7:4] — Miscellaneous Commands
The encoded value of this field may be used to specify a single command as follows:

0000	No command.
0001	Reset MR pointer. Causes the MR pointer to point to MR1.
0010	Reset receiver. Resets the receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO pointer is reset to the first location.
0011	Reset transmitter. Resets the transmitter as if a hardware reset had been applied.
0100	Reset error status. Clears the received break, parity error, framing error, and overrun error bits in the status register (SR[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared), and in block mode to clear all error status after a block of data has been received.
0101	Reset break change interrupt. Causes the break detect change bit in the interrupt status register (ISR[2 or 6]) to be cleared to zero.
0110	Start break. Forces the TxD output low (spacing). If the transmitter is empty, the start of the break condition will be delayed up to two bit times. If the transmitter is active, the break begins when transmission of the character is completed. If a character is in the THR, the start of break is delayed until that character or any others loaded after it have been transmitted (TxEMT must be true before break begins). The transmitter must be enabled to start a break.
0111	Stop break. The TxD line will go high (marking) within two bit times. TxD will remain high for one bit time before the next character, if any, is transmitted.
1000	Assert RTSN. Causes the RTSN output to be asserted (low).
1001	Negate RTSN. Causes the RTSN output to be negated (high).
1010	Set special time out mode with this channel as the channel to restart the C/T as each receive character is transferred from shift register to RHR.
1011	Reserved.
1100	Reset special time out mode.
1101	Reserved.
111x	Reserved for testing.

CR[3] — Disable Transmitter
This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CR[2] — Enable Transmitter
Enables operation of the channel a transmitter. The TxRDY status bit will be asserted.

CR[1] — Disable Receiver
This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special wake-up mode is programmed, the receiver operates even if it is disabled (see Wake-Up Mode).

CR[0] — Enable Receiver
Enables operation of the receiver. If not in the special wake-up mode, this also forces the receiver into the search for start bit state.

Table 3. Baud Rate

CSR[7:4]	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	38.4k
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4k	19.2k
1 1 0 1	Timer	Timer
1 1 1 0	MPI3 — 16×	MPI3 — 16×
1 1 1 1	MPI3 — 1×	MPI3 — 1×

SR — Channel Status Register

SR[7] — Received Break
This bit indicates that an all-zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received; further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half bit time (two successive edges of the internal or external 1× clock).

When this bit is set, the change in break bit in the ISR (ISR[6 or 2]) is set. ISR[6 or 2] is also set when the end of the break condition, as defined above, is detected. The break detect circuitry is capable of detecting breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must last until the end of the next character in order for it to be detected.

SR[6] — Framing Error (FE)
This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SR[5] — Parity Error (PE)
This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity. In the special 'wake-up' mode, the parity error bit stores the received A/D bit.

SR[4] — Overrun Error (OE)
This bit, when set, indicates that one or more characters in the received data stream have

been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost. This bit is cleared by a reset error status command.

SR[3] — Transmitter Empty (TxEMT)
This bit is set when the transmitter underruns, i.e., both the transmit holding register and the transmit shift register are empty. However, this bit is not set until one character has been transmitted. It is set after transmission of the last stop bit of a character, if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU, or when the transmitter is disabled.

SR[2] — Transmitter Ready (TxRDY)
This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, e.g., characters loaded in the THR while the transmitter is disabled will not be transmitted.

SR[1] — FIFO Full (FFULL)
This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the FIFO and there is no character in the receive shift register. If a character is waiting in the receive shift register because the FIFO is full, FFULL is not reset after reading the FIFO once.

SR[0] — Receiver Ready (RxRDY)
This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, and no more characters are in the FIFO.

OPCR — Output Port Configuration Register

OPCR[6:4] — MPOb Output Select
This field programs the MPOb output pin to provide one of the following:

000 Request-to-send active low output (RTSN). This output is asserted and negated via the command register. Mode RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted out or when the receiver FIFO and receiver shift register are full using MR2[5] and MR1[7], respectively.

001 The counter/timer output. In the timer mode, this output is a square wave with a period of twice the value (in clock periods) of the contents of the CTUR and CTLR. In the counter mode, the output remains high until the terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command.

010 The 1× clock for the transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a non-synchronized 1× clock is output.

011 The 16× clock for the transmitter. This is the clock selected by CSR[3:0], and is a 1× clock if CSR[3:0] = 1111.

100 The 1× clock for the receiver, which is the clock that samples the received data. If data is not being received, a non-synchronized 1× clock is output.

101 The 16× clock for the receiver. This is the clock selected by CSR[7:4], and is a 1× clock if CSR[7:4] = 1111.

110 The transmitter register empty signal, which is the same as SR[3].

111 The receiver ready or FIFO full signal.

OPCR[3] — Power-Down Mode Select
This bit, when set, selects the power-down mode. In this mode, the SCC2698 oscillator is stopped and all functions requiring this clock are suspended. The contents of all registers are saved. It is recommended that the transmitter and receiver be disabled prior to placing the SCC2698 in this mode. This bit is reset with RESET asserted. Note that this bit must be set to a logic 1 before power-down occurs. Only OPCR[3] in block A controls the power-down mode.

OPCR[2:0] — MPOa Output Select
This field programs the MPOa output pin to provide one of the same functions as described in OPCR[6:4].

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select
This bit selects one of two sets of baud rates to be generated by the BRG.

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05k, 1.2k, 2.4k, 4.8k, 7.2k, 9.6k, and 38.4k baud.

Set 2: 75, 110, 38.4k, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k and 19.2k baud.

The selected set of rates is available for use by the receiver and transmitter.

ACR[6:4] — Counter/Timer Mode and Clock Source Select
This field selects the operating mode of the counter/timer and its clock source (see Table 4).

The MPI pin available as the clock source is MPI a, c, e, and g only.

ACR[3:0] — MPI1b, MPI0b, MPI1a, MPI0a Change-of-State Interrupt Enable
This field selects which bits of the input port change register (IPCR) cause the input change bit in the interrupt status register, ISR[7], to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — MPI1b, MPI0b, MPI1a, MPI0a Change-of-State
These bits are set when a change-of-state, as defined in the Input Port section of this data sheet, occurs at the respective pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register. The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — MPI1b, MPI0b, MPI1a, MPI0a Change-of-State
These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

Table 4. ACR[6:4] Operating Mode

[6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	MPI pin
0 0 1	Counter	MPI pin divided by 16
0 1 0	Counter	TxC — 1× clock of the transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	MPI pin
1 0 1	Timer	MPI pin divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a "1" and the corresponding bit in the IMR is also a '1', the INTRN output is asserted (low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status is provided regardless of the contents of the IMR.

ISR[7] — MPI Change-of-State
This bit is set when a change-of-state occurs at the MPI1b, MPI0b, MPI1a, MPI0a input pins. It is reset when the CPU reads the IPCR.

ISR[6] — Channel b Change in Break
This bit, when set, indicates that the receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a reset break change interrupt command.

ISR[5] — Receiver Ready or FIFO Full Channel b
The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the receive FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read.

If programmed as FIFO full, it is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when FIFO is read and there is no character in the receive shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

ISR[4] — Transmitter Ready Channel b
This bit is a duplicate of TxRDY (SR[2]).

ISR[3] — Counter Ready
In the counter mode of operation, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command. It is initialized to '0' when the chip is reset.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time the C/T reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the C/T.

ISR[2] — Channel a Change in Break
This bit, when set, indicates that the receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a reset break change interrupt command.

ISR[1] — Receiver Ready or FIFO Full Channel a
The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the receive FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read. If programmed as FIFO full, it is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when FIFO is read and there is no character in the receiver shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

ISR[0] — Transmitter Ready Channel a
This bit is a duplicate of TxRDY (SR[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is a '1', the INTRN output is asserted (low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no affect on the INTRN output. Note that the IMR does not mask reading of the ISR.

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs, respectively, of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded is 0002_{16}.

In the timer (programmable-divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half-periods will be. In this mode the C/T runs continuously. Receipt of a start counter command causes the counter to begin a new cycle using the values in CTU and CTL. The counter ready status bit, ISR[3], is set once each cycle of the square wave. The bit is reset by a stop counter command. The command, however, does not stop the C/T.

The generated square wave is output on MPO if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR. Counting begins upon receipt of a start counter command. Upon reaching the terminal count, the counter ready interrupt bit, ISR[3], is set. The counter continues counting past the terminal count until stopped by the CPU. If MPO is programmed to be the output of the C/T, the output remains high until the terminal count is reached, at which time it goes low.

The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower eight bits of the counter can be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower eight bits to the upper eight bits occurs between the times that both halves of the counter are read. However, a subsequent start counter command causes the counter to begin a new count cycle using the values in CTUR and CTLR.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	-65 to +150	°C
V_{CC}	Voltage from V_{CC} to GND[3]	-0.5 to +7.0	V
V_S	Voltage from any pin to ground[3]	-0.5 to V_{CC} ±5%	V
P_D	Power dissipation	1	W

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2			V
V_{IH}	Input high voltage (X1/CLK)		0.9V_{CC}	V_{CC}		V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except OC outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	-10		10	μA
I_{LL}	Data bus 3-State leakage current	$V_O = 0$ to V_{CC}	-10		10	μA
I_{X1L}	X1/CLK low input current	$V_{IN} = 0$, X2 floated	-3	-1.5	0	mA
I_{X1H}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 floated	0	3.5	10	mA
I_{X2L}	X2 low input current	$V_{IN} = 0$, X1/CLK floated	-100	-30	0	μA
I_{X2H}	X2 high input current	$V_{IN} = V_{CC}$, X1/CLK floated	0	+30	100	μA
I_{OC}	Open-collector output leakage current	$V_O = 0$ to V_{CC}			10	μA
I_{CC}	Power supply current				150	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all inputs except X1/CLK swing between 0.4V and 2.4V with a transition time of 20ns maximum. For X1/CLK this swing is between 0.4V and 4.4V. All time measurements are referenced at input voltages of 0.8V and 2V, as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.

Octal Universal Asynchronous Receiver/Transmitter (Octal-UART)

SCC2698

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$[4, 5, 6, 7]

SYMBOL	PARAMETER	TENTATIVE LIMITS			UNIT
		Min	Typ	Max	
Reset timing (Figure 2)					
t_{RES}	RESET pulse width		1.0		μs
Bus timing (Figure 3)[8]					
t_{AS}	A0 – A5 setup time to RDN, WRN low	10			ns
t_{AH}	A0 – A5 hold time from RDN, WRN high	0			ns
t_{CS}[9]	CEN setup time to RDN, WRN low	0			ns
t_{CH}[9]	CEN hold time from RDN, WRN high	0			ns
t_{RW}	WRN, RDN pulse width	225			ns
t_{DD}	Data valid after RDN low			150	ns
t_{DF}	Data bus floating after RDN high			100	ns
t_{DS}	Data setup time before WRN high	100			ns
t_{DH}	Data hold time after WRN high	10			ns
t_{RWD}[10]	High time between reads and/or writes	200			ns
MPI and MPO timing (Figure 4)[8]					
t_{PS}	MPI input setup time before RDN low	0			ns
t_{PH}	MPI input hold time after RDN high	0			ns
t_{PD}	MPO output valid after WRN high			370	ns
Interrupt timing (Figure 5)					
t_{IR}	INTRN high from:				
	Read RHR (RxRDY/FFULL interrupt)			270	ns
	Write THR (TxRDY, TxEMT interrupt)			270	ns
	Reset command (break change interrupt)			270	ns
	Reset command (MPI change interrupt)			270	ns
	Stop C/T command (counter interrupt)			270	ns
	Write IMR (clear of interrupt mask bit)			270	ns
Clock timing (Figure 6)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2.0	3.6864	4	MHz
t_{CTC}	CTCLK high or low time	100			ns
f_{CTC}	CTCLK frequency	0		4	MHz
t_{RX}	RxC high or low time	220			ns
f_{RX}	RxC frequency (16×)	0		2	MHz
	(1×)	0		1	MHz
t_{TX}	TxC high or low time	220			ns
f_{TX}	TxC frequency (16×)	0		2	MHz
	(1×)	0		1	MHz
Transmitter timing (Figure 7)					
t_{TXD}	TxD output delay from TxC low			350	ns
t_{TCS}	TxC output delay from TxD output data	0		150	ns
Receiver timing (Figure 8)					
t_{RXS}	RxD data setup time to RxC high	200			ns
t_{RXH}	RxD data hold time from RxC high	200			ns

NOTES:

7. Test condition for outputs: $C_L = 150pF$, except interrupt outputs. Test condition for interrupt outputs: $C_L = 50pF$, $R_L = 2.7k\Omega$ to V_{CC}.
8. Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the 'strobing' input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN. CEN and RDN (also CEN and WRN) are ANDed internally. As a consequence, the signal asserted last initiates the cycle and the signal negated first terminates the cycle.
9. If CEN is used as the 'strobing' input, the parameter defines the minimum high times between one CEN and the next. The RDN signal must be negated for t_{RWD} to guarantee that any status register changes are valid.
10. Consecutive write operations to the same command register require at least three edges of the X1 clock between writes.

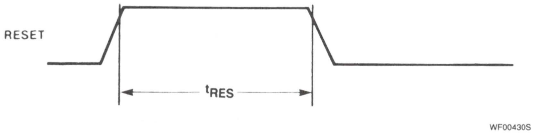

RESET
t_{RES}
WF00430S

Figure 2. Reset Timing

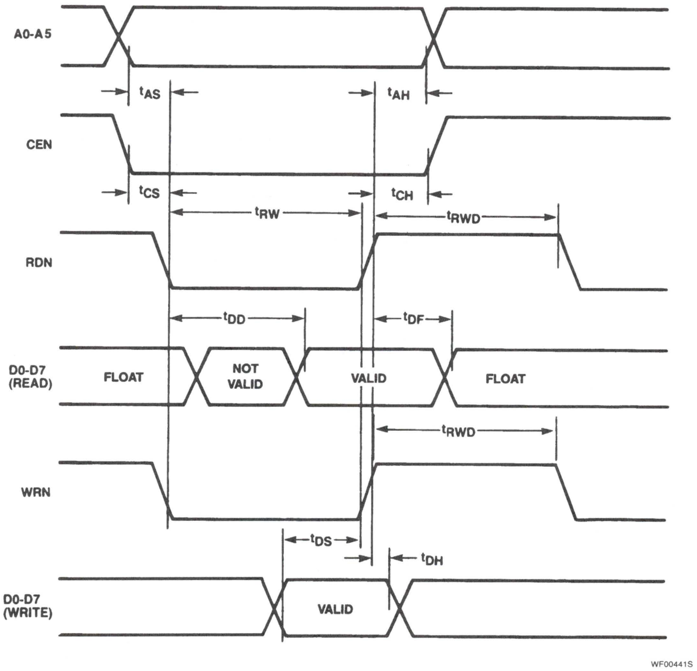

A0-A5
t_{AS}
t_{AH}
CEN
t_{CS}
t_{CH}
t_{RW}
t_{RWD}
RDN
t_{DD}
t_{DF}
D0-D7
(READ)
FLOAT
NOT VALID
VALID
FLOAT
t_{RWD}
WRN
t_{DS}
t_{DH}
D0-D7
(WRITE)
VALID
WF00441S

Figure 3. Bus Timing

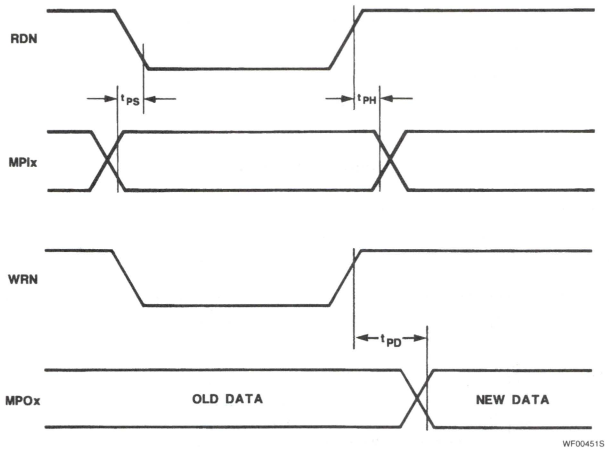
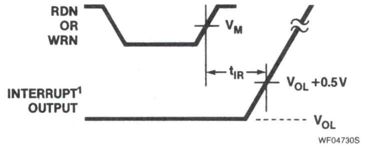

RDN
tPS
tPH
MPIx
WRN
tPD
MPOx
OLD DATA
NEW DATA
WF00451S

Figure 4. Port Timing


RDN OR WRN
VM
tIR
VOL +0.5V
INTERRUPT1 OUTPUT
VOL
WF04730S

NOTES:

1. INTRN or MPO when used as interrupt outputs.
2. The test for open drain outputs is intended to guarantee switching of the output transistor. Measurement of this response is referenced from the midpoint of the switching signal, V_M, to a point 0.5V above V_{OL}. This point represents noise margin that assures true switching has occurred. Beyond this level, the effects of external circuitry and test environment are pronounced and can greatly affect the resultant measurement.

Figure 5. Interrupt Timing

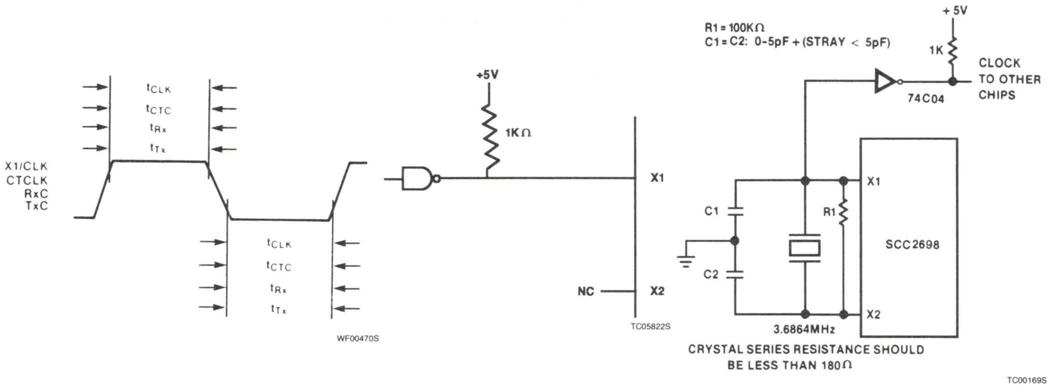

tCLK
tCTC
tRx
tTx
X1/CLK
CTCLK
RxC
TxC
tCLK
tCTC
tRx
tTx
WF00470S
+5V
1KΩ
X1
NC
X2
TC05822S
R1 = 100KΩ
C1 = C2: 0-5pF + (STRAY < 5pF)
+5V
1K
CLOCK TO OTHER CHIPS
74C04
X1
C1
R1
SCC2698
C2
X2
3.6864MHz
CRYSTAL SERIES RESISTANCE SHOULD BE LESS THAN 180Ω
TC00169S

Figure 6. Clock Timing

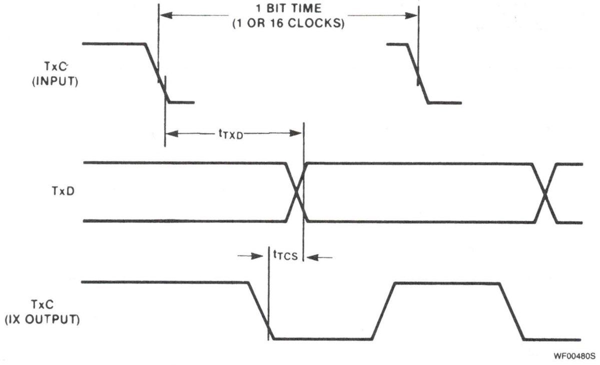

1 BIT TIME
(1 OR 16 CLOCKS)
TxC
(INPUT)
tTXD
TxD
tTCS
TxC
(IX OUTPUT)
WF00480S

Figure 7. Transmit

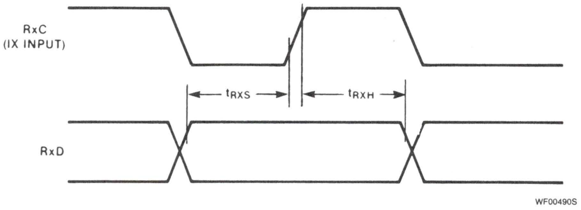
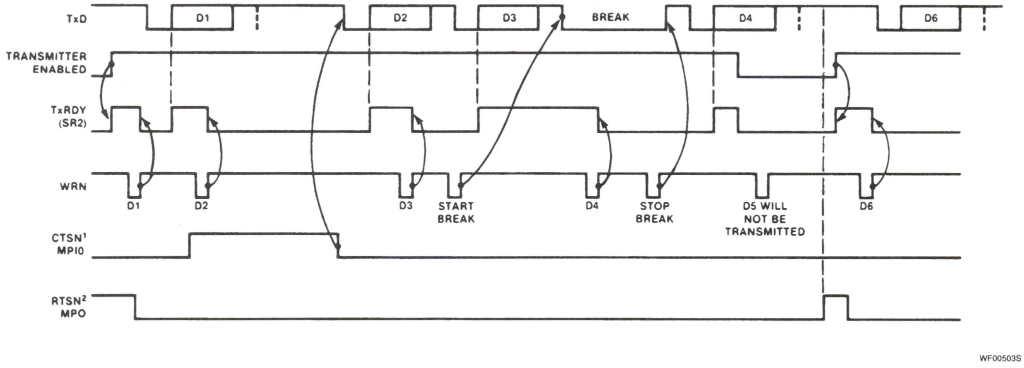
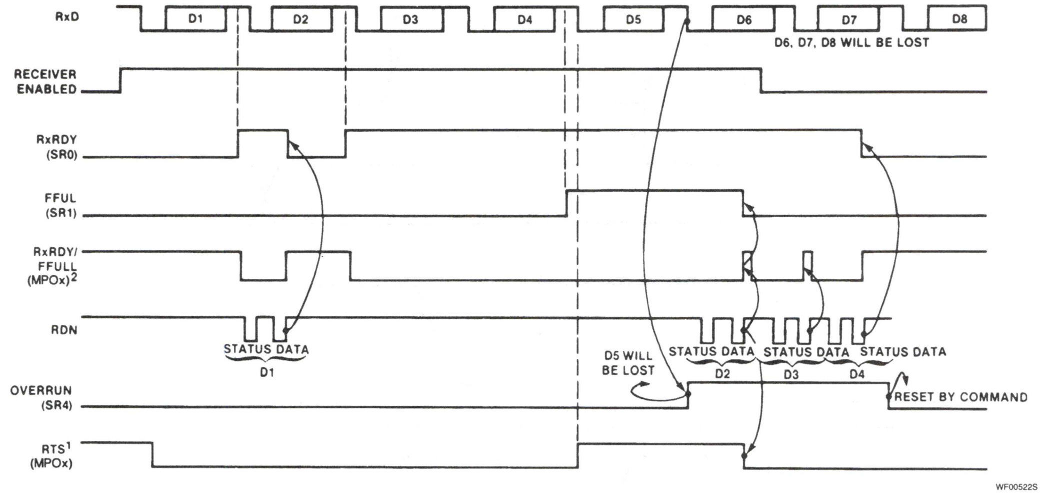

RxC
(IX INPUT)
tRXS
tRXH
RxD
WF00490S

Figure 8. Receive


TxD
D1
D2
D3
BREAK
D4
D6
TRANSMITTER
ENABLED
TxRDY
(SR2)
WRN
D1
D2
D3
START
BREAK
D4
STOP
BREAK
D5 WILL
NOT BE
TRANSMITTED
D6
CTSN[1]
MPIO
RTSN[2]
MPO
WF00503S

NOTES:
1. Timing shown for MR2[4] = 1.
2. Timing shown for MR2[5] = 1.

Figure 9. Transmitter Timing


RxD
D1
D2
D3
D4
D5
D6
D7
D8
D6, D7, D8 WILL BE LOST
RECEIVER ENABLED
RxRDY (SR0)
FFUL (SR1)
RxRDY/ FFULL (MPOx)2
RDN
STATUS DATA
D1
D5 WILL BE LOST
STATUS DATA STATUS DATA STATUS DATA
D2 D3 D4
OVERRUN (SR4)
RESET BY COMMAND
RTS1 (MPOx)
WF00522S

NOTES:
1. Timing shown for MR1[7] = 1.
2. Shown for OPCR[6:4, 2:0] = 111 or 000 and MR1[6] = 0.

Figure 10. Receiver Timing

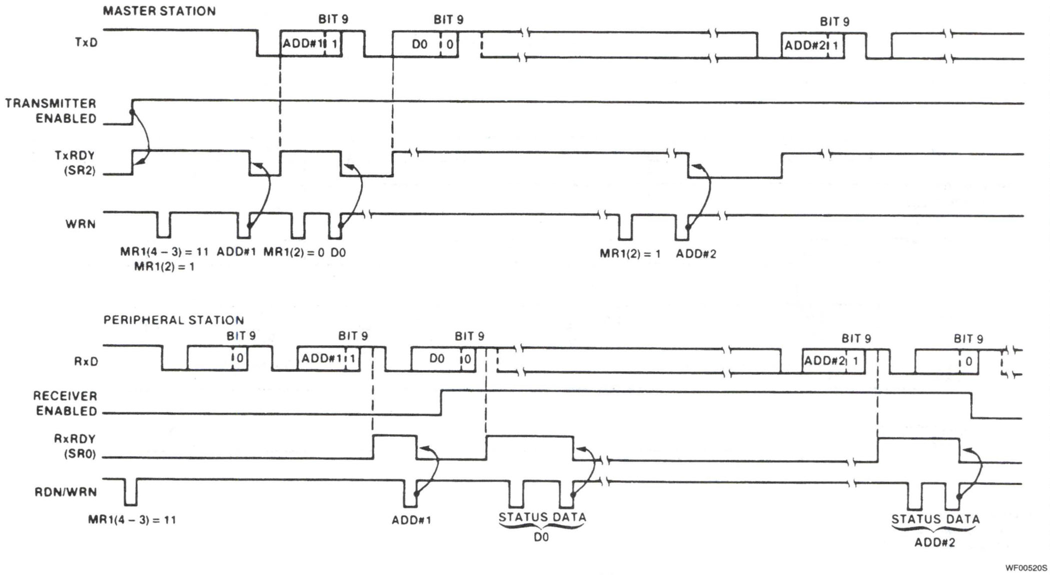

MASTER STATION
TxD
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
TRANSMITTER ENABLED
TxRDY (SR2)
WRN
MR1(4 – 3) = 11 ADD#1 MR1(2) = 0 D0
MR1(2) = 1
MR1(2) = 1 ADD#2
PERIPHERAL STATION
RxD
BIT 9
0
BIT 9
ADD#1 1
BIT 9
D0 0
BIT 9
ADD#2 1
BIT 9
0
RECEIVER ENABLED
RxRDY (SR0)
RDN/WRN
MR1(4 – 3) = 11
ADD#1
STATUS DATA
D0
STATUS DATA
ADD#2
WF00520S

Figure 11. Wake-Up Mode

Signetics

SCC63484
Advanced CRT Controller (ACRTC)

Preliminary Specification

Microprocessor Products

DESCRIPTION

The Signetics SCC63484 Advanced CRT Controller (ACRTC) is a CMOS VLSI microcomputer peripheral device capable of controlling raster scan type CRTs to display both graphics and characters. The ACRTC is a new generation CRT controller that is based on a bit-mapped technology and has more display control functions than those of an SCN2674 Advanced Video Display Controller (AVDC).

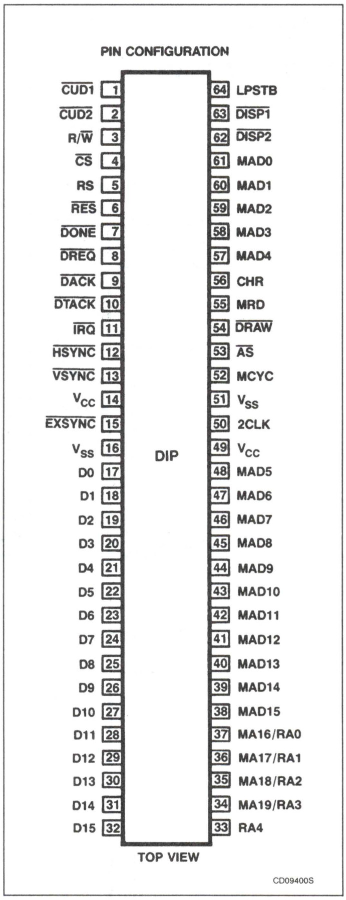
The ACRTC prepares the mechanisms to use in one of three modes: character only; graphic only and multiplexed character/graphic modes. Therefore, the ACRTC can be applied to many applications, from character-only display devices to large full-graphic systems.

The ACRTC can reduce CPU software overhead and enhance system throughput.

FEATURES

- **High-speed graphic drawings**
  - **Drawing rate: maximum 500ns/pixel (color drawing)**
  - **Drawn graphics: Dot, line, rectangle, polyline, polygon, circle, ellipse, paint, copy, etc.**
  - **Drawn colors: 16 bits/word, 1, 2, 4, 8, 16 bits/pixel (5 types) monochrome to max 64k colors**
- **Large frame memory space**
  - **Maximum 2Mbytes graphic memory**
  - **Maximum 128k-byte character memory separated from the MPU memory**
  - **Available to maximum 4096 × 4096 high-resolution CRT (1 bit/pixel mode)**
- **Various CRT display controls**
  - **Split screens (3 displays and 1 window)**
  - **Zooming up (1 to 16 times)**
  - **Scroll (vertical and horizontal)**
- **External synchronization**
  - **Synchronization between ACRTCs or between the ACRTC and external device; e.g. TV system or other controller**
- **DMA interface**
- **Two programmable cursors**
- **Three scan modes**
  - **Non-interlace, interlace sync., and interlace sync. and video modes**
- **Interrupt request to MPU**
- **256 characters/line, 32 rasters/line, 4096 rasters/screen**
- **Maximum clock frequency 8MHz**
- **CMOS, +5V single power supply**

PIN CONFIGURATION

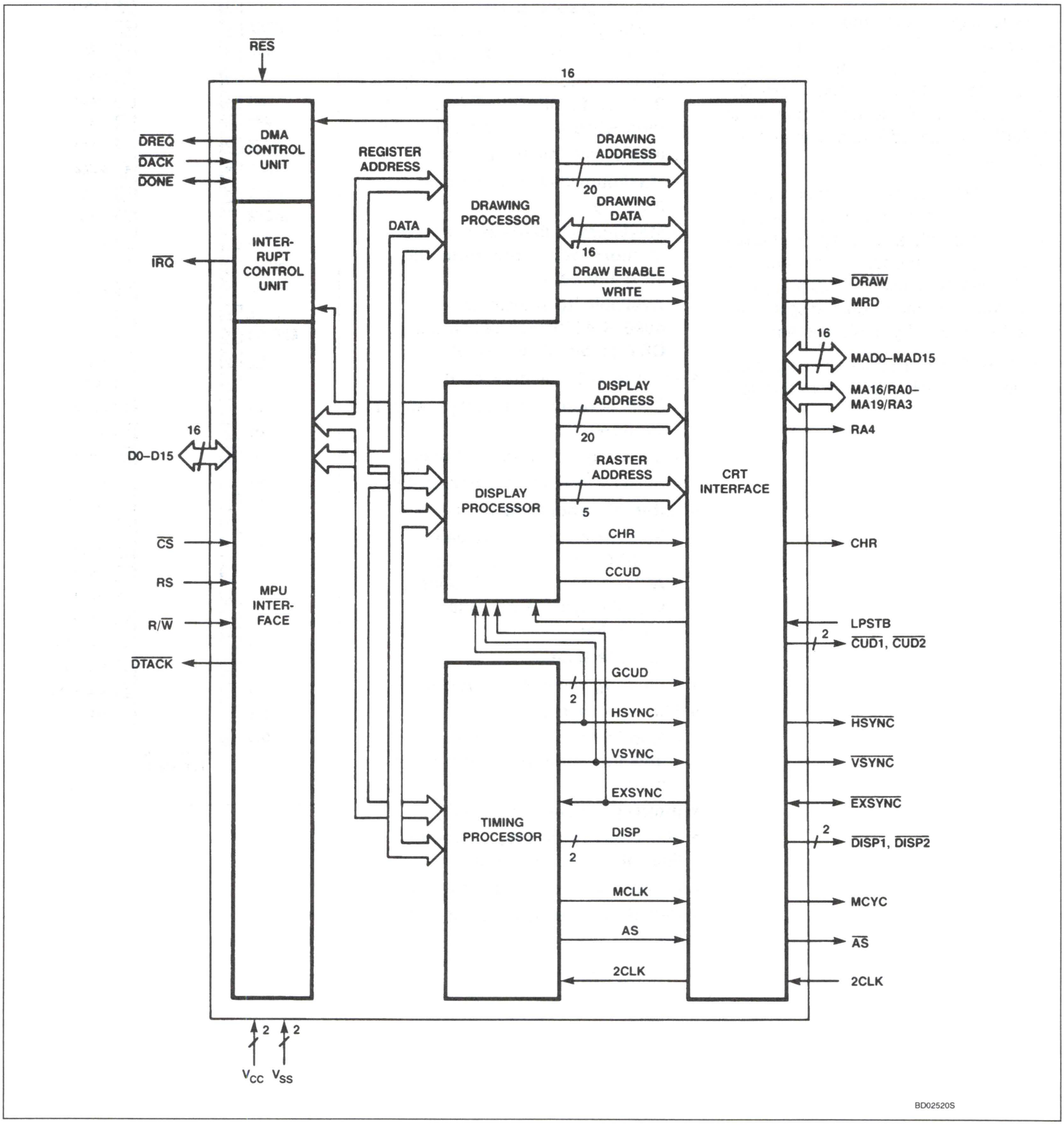

PIN CONFIGURATION
$\overline{CUD1}$ 1
$\overline{CUD2}$ 2
R/$\overline{W}$ 3
$\overline{CS}$ 4
RS 5
$\overline{RES}$ 6
$\overline{DONE}$ 7
$\overline{DREQ}$ 8
$\overline{DACK}$ 9
$\overline{DTACK}$ 10
$\overline{IRQ}$ 11
$\overline{HSYNC}$ 12
$\overline{VSYNC}$ 13
V_{CC} 14
$\overline{EXSYNC}$ 15
V_{SS} 16
D0 17
D1 18
D2 19
D3 20
D4 21
D5 22
D6 23
D7 24
D8 25
D9 26
D10 27
D11 28
D12 29
D13 30
D14 31
D15 32
64 LPSTB
63 $\overline{DISP1}$
62 $\overline{DISP2}$
61 MAD0
60 MAD1
59 MAD2
58 MAD3
57 MAD4
56 CHR
55 MRD
54 $\overline{DRAW}$
53 $\overline{AS}$
52 MCYC
51 V_{SS}
50 2CLK
49 V_{CC}
48 MAD5
47 MAD6
46 MAD7
45 MAD8
44 MAD9
43 MAD10
42 MAD11
41 MAD12
40 MAD13
39 MAD14
38 MAD15
37 MA16/RA0
36 MA17/RA1
35 MA18/RA2
34 MA19/RA3
33 RA4
DIP
TOP VIEW
CD09400S

Advanced CRT Controller (ACRTC) SCC63484

ORDERING INFORMATION

PACKAGE	V_{CC} = 5V ±5%, T_A = 0°C to +70°C
	8MHz
Plastic DIP	SCC63484C8N64
Ceramic DIP	SCC63484C8I64
Plastic LCC	SCC63484C8A68

BLOCK DIAGRAM


RES
16
DREQ
DACK
DONE
DMA CONTROL UNIT
REGISTER ADDRESS
DRAWING PROCESSOR
DATA
DRAWING ADDRESS
20
DRAWING DATA
16
DRAW ENABLE
WRITE
IRQ
INTER-RUPT CONTROL UNIT
DRAW
MRD
16
MAD0–MAD15
MA16/RA0–MA19/RA3
RA4
DISPLAY ADDRESS
20
RASTER ADDRESS
5
DISPLAY PROCESSOR
CRT INTERFACE
16
D0–D15
CHR
CCUD
CHR
CS
RS
R/W
DTACK
MPU INTER-FACE
LPSTB
2
CUD1, CUD2
GCUD
2
HSYNC
VSYNC
EXSYNC
HSYNC
VSYNC
EXSYNC
TIMING PROCESSOR
DISP
2
2
DISP1, DISP2
MCLK
MCYC
AS
AS
2CLK
2CLK
2
2
V_{CC} V_{SS}
BD02520S

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
MPU interface			
$\overline{RES}$	6	I	**Reset:** Hardware reset to the ACRTC.
D0 – D15	17 – 32	I/O	**Data Bus:** The bidirectional data bus for communication with the host MPU or DMAC. In 8-bit data bus mode, D0 – D7 are used.
R/$\overline{W}$	3	I	**Read/Write:** Controls the direction of host to/from ACRTC transfers.
$\overline{CS}$	4	I	**Chip Select:** Enables data transfers between the host and the ACRTC.
RS	5	I	**Register Select:** Selects the ACRTC register to be accessed and is normally connected to the least significant bit of the host address bus.
$\overline{DTACK}$	10	O	**Data Transfer Acknowledge:** Provides asynchronous bus cycle timing and is compatible with the SCN68000 Microprocessor $\overline{DTACK}$ input.
$\overline{IRQ}$	11	O	**Interrupt Request:** Generates interrupt service requests to the host MPU.
DMAC interface			
$\overline{DREQ}$	8	O	**DMA Request:** Generates DMA service requests to the host DMAC.
$\overline{DACK}$	9	I	**DMA Acknowledge:** Receives DMA acknowledge timing from the host DMAC.
$\overline{DONE}$	7	I/O	**Done:** Terminates DMA transfer and is compatible with the DMAC $\overline{DONE}$ signal.
CRT interface			
2CLK	50	I	**Clock:** Basic ACRTC operating clock derived from the dot clock.
MAD0 – MAD15	61 – 57, 48 – 38	I/O	**Address/Data Bus:** Multiplexed frame buffer address/data bus.
$\overline{AS}$	53	O	**Address Strobe:** Address strobe for demultiplexing the frame buffer address/data bus (MAD0 – MAD15).
MA16/RA0 – MA19/RA3	37 – 34	O	**Address Bits/Raster Address Outputs:** The high-order address bits for graphic screens and the raster address outputs for character screens.
RA4	33	O	**Raster Address Bit:** Provides the high-order raster address bit (up to 32 rasters) for character screens.
CHR	56	O	**Graphic/Character Screen Access:** Indicates whether a graphic or character screen is being accessed.
MCYC	52	O	**Memory Clock:** Frame buffer memory access timing — one half the frequency of 2CLK.
MRD	55	O	**Bus Direction Control:** Frame buffer data bus direction control.
$\overline{DRAW}$	54	O	**Drawing/Refresh Cycle:** Differentiates between drawing cycles and CRT display refresh cycles.
$\overline{DISP1}$ – $\overline{DISP2}$	62 – 63	O	**Display Enable Timing:** Programmable display enable timing used to selectively enable, disable and blank logical screens.
$\overline{CUD1}$ – $\overline{CUD2}$	1 – 2	O	**Cursor Timing:** Provides cursor timing determined by ACRTC programmed parameters such as cursor definition, cursor mode, cursor address, etc.
$\overline{VSYNC}$	13	O	**Vertical Synchronization:** CRT device vertical synchronization pulse.
$\overline{HSYNC}$	12	O	**Horizontal Synchronization:** CRT device horizontal synchronization pulse.
$\overline{EXSYNC}$	15	I/O	**External Synchronization:** For synchronization between multiple ACRTCs and other video signal generating devices.
LPSTB	64	I	**Light Pen:** Connection to an external light pen.

BLOCK DIAGRAM

The ACRTC consists of five major functional blocks. These functional blocks operate in parallel to achieve maximum performance. Two of the blocks perform the external bus interface for the host MPU and CRT, respectively.

MPU Interface

The MPU interface manages the asynchronous host MPU interface including the programmable interrupt control unit and DMA handshaking control unit.

CRT Interface

The CRT interface manages the frame buffer bus and CRT timing input and output control signals. Also, the selection of either display refresh address or drawing address outputs is performed. The other three blocks are separately microprogrammed processors which operate in parallel to perform the major functions of drawing, display control, and timing.

Drawing Processor

This interprets commands and command parameters issued by the host bus (MPU and/or DMAC) and performs the drawing operations on the frame buffer memory. This processor is responsible for the execution of ACRTC drawing algorithms and conversion of logical pixel X-Y addresses to physical frame buffer addresses. Communication with the host bus is from separate 16-byte read and write FIFOs.

Display Processor

The display processor manages frame buffer refresh addressing based on the user-programmed specification of display screen organization. Combines and displays as many as four independent screen segments (three horizontal splits and one window) using an internal high-speed address calculation unit. Controls display refresh address outputs based on graphic (physical frame buffer address) or character (physical frame buffer address + row address) display modes.

Timing Processor

This generates the CRT synchronization signals and other timing signals used internally by the ACRTC. The ACRTC's software visible registers are similarly partitioned and reside in the appropriate internal processor, depending on function. The registers in the display and timing processors are loaded with basic display parameters during system initialization. During operation, the host primarily communicates with the ACRTC's drawing processor via the on-chip FIFOs.

1. High-level command language increases performance and reduces software development cost.
   - ACRTC converts logical X-Y coordinates to physical frame buffer addresses
   - 38 commands including 23 graphic drawing commands — LINE, RECTANGLE, POLYLINE, POLYGON, CIRCLE, ELLIPSE, ARC, ELLIPSE ARC, FILLED RECTANGLE, PAINT, PATTERN and COPY
   - On-chip 32-byte pattern RAM
   - Conditional drawing function (8 conditions) for drawing patterns, color mixing and software mixing and software windowing
   - Drawing area control with hardware clipping and hitting
   - Maximum drawing speed of 2 million logical pixels per second is the same for monochrome and color applications
2. High resolution display with advanced screen control
   - Up to 4096 by 4096 bit map graphic display and/or 256 line by 256 character by 32 raster character display
   - Separate bit map graphic (2Mbyte) and character (128kbyte) address spaces with combined graphic/character display
   - Three horizontal split screens and one window screen
   - Size and position fully programmable
   - Independent horizontal and vertical smooth scroll for each screen
   - 1 to 16 zoom magnitude — independent X and Y zoom factors
   - Logical pixel specification as 1, 2, 4, 8 or 16 bits for monochrome, gray scale and color displays
   - Programmable address increment supports frame buffer memory widths to 256 bits for video bit rates > 500MHz. (ACRTC R mask is limited to 128 bits)
   - Unique interleaved access mode for screen superimposition or 'flashless' displays
   - ACRTC provides dynamic RAM refresh address
3. High-performance MPU interface
   - Optimized interface with the SCN68000 MPU and DMAC
   - 8- or 16-bit bus — compatible with other MPUs
   - Separate on-chip 16-byte read and write FIFOs
   - Maskable interrupts including FIFO status
4. Versatile CRT interface
   - Full programmability of CRT timing signals
   - Three raster scanning modes
   - Master or slave synchronization to multiple ACRTCs or other video generating devices
   - Two hardware cursors; three cursor modes
   - Programmable cursor and display timing skew
   - Eight user-defineable video attributes
   - Light pen detection
5. VLSI CMOS process

OPERATION

Powerful visual interfaces are a key component of advanced system architectures. A proven technique uses raster-scanned CRT technology for the display of graphics and text information.

Systems which use first-generation CRT controllers (CRTCs) are constrained by hardware/software design time, manufacturing cost, and limited MPU bandwidth. To meet the functional requirements for powerful visual interfaces and to support their use in high volume, cost-sensitive applications, advanced circuit design and VLSI CMOS manufacturing technologies have been used to create a next generation CRTC, the SCC63484 ACRTC. The ACRTC concept is to incorporate major functionality on-chip, formerly requiring external hardware and software. In this way, both higher performance and reduced system cost benefits are achieved.

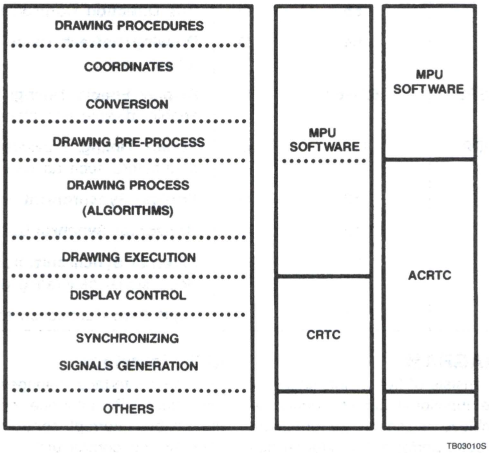


Figure 1. ACRTC vs CRTC

Advanced CRT Controller (ACRTC) SCC63484

APPLICATIONS

The overall function of a visual interface is logically partitioned into layers. At the lowest layer are CRT timing and control signal generation. At the top layer are general purpose drawing procedures which provide a high-level interface to the user's operating system or application software. At this layer, a number of popular standards have emerged including GKS, CORE, NAPLP, GSX and others.

Figure 1 shows how the ACRTC performs the key functions or logical drawing algorithm and physical drawing execution. Formerly, these functions were performed by external hardware and/or MPU software.

As shown, the ACRTC reduces the 'gap' between device functionality and high-level graphics procedures. Since the ACRTC device itself provides capabilities closely related to those of high-level graphics packages, the effort (hardware and software design time and cost) required to develop a visual interface is significantly reduced.

Noting the traditional and emerging applications for visual interfaces, Figure 2 shows that a single ACRTC is suitable for a broad range of products in both alphanumeric and graphics areas. Multiple ACRTCs can achieve performance beyond that of any first-generation CRTC configuration.

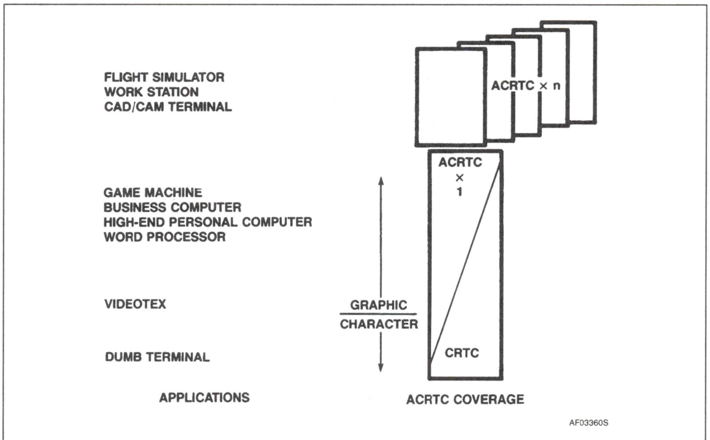

FLIGHT SIMULATOR
WORK STATION
CAD/CAM TERMINAL
ACRTC × n
GAME MACHINE
BUSINESS COMPUTER
HIGH-END PERSONAL COMPUTER
WORD PROCESSOR
ACRTC × 1
VIDEOTEX
GRAPHIC
CHARACTER
DUMB TERMINAL
CRTC
APPLICATIONS
ACRTC COVERAGE
AF03360S

Figure 2. Application Spectrum

SYSTEM CONFIGURATION

Existing CRTCs provide a single bus interface to the frame buffer which must be shared with the host MPU. However, the refresh of large frame buffers and the requirement to access the frame buffer for drawing operations can quickly saturate this shared bus bandwidth.

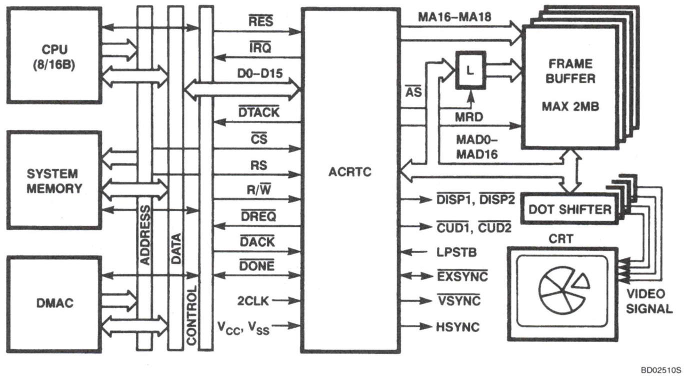

CPU (8/16B)
SYSTEM MEMORY
DMAC
ADDRESS
DATA
CONTROL
RES
IRQ
D0–D15
DTACK
CS
RS
R/W
DREQ
DACK
DONE
2CLK
VCC, VSS
ACRTC
MA16–MA18
AS
L
MRD
MAD0–MAD16
FRAME BUFFER
MAX 2MB
DISP1, DISP2
CUD1, CUD2
LPSTB
EXSYNC
VSYNC
HSYNC
DOT SHIFTER
CRT
VIDEO SIGNAL
BD02510S

Figure 3. System Configuration

As shown in Figure 3, the ACRTC uses separate host MPU and frame buffer bus interfaces. This allows the ACRTC full access to the frame buffer for display refresh, DRAM refresh and drawing operations while minimizing the ACRTC's usage of the MPU system bus. Thus, overall system performance is maximized. A related benefit is that a large frame buffer (2Mbyte for each ACRTC) is usable even if the host MPU has a smaller address space or segment size restriction.

The ACRTC can utilize an external DMA controller. This increases system throughput when large amounts of command, parameter and data information must be transferred to the ACRTC. Also, advanced DMAC features, such as the 'chaining' modes, can be used to develop powerful graphics system architectures.

However, more cost-sensitive or less performance-sensitive applications do not require a DMAC. The interface to the ACRTC can be handled completely under MPU software control.

While both ACRTC bus interfaces (host MPU and frame buffer) exploit 16-bit data paths for maximum performance, the ACRTC also offers an 8-bit MPU mode for easy connection to popular 8-bit bus structures.

VIDEO ATTRIBUTES

The ACRTC outputs 20 bits of video attributes on MAD0 – MAD15 and MA16/RA0 – MA19/RA3. These attributes are output at the last cycle prior to the rising edge of HSYNC and should be latched externally. Thus, video attributes can be set on a raster-by-raster basis (see Figure 4).

Attribute Code (ATC0 – ATC7:MAD0 – MAD7)

These are user-defined attributes. The programmed contents of the attribute control bits (ATR) of the display control register (DCR) are output on these lines. Note that the data written into ATR can be externally used after the completion of current raster scanning.

Attribute Code (ATC7 – ATC0) Application

ATC is one of the functions provided for user applications; the data employed depends on the system requirements. Application selections include:

1. Amount of horizontal dot shift for window smooth scroll
2. Horizontal width of crosshair cursor and the amount of horizontal dot shift (including block cursor)

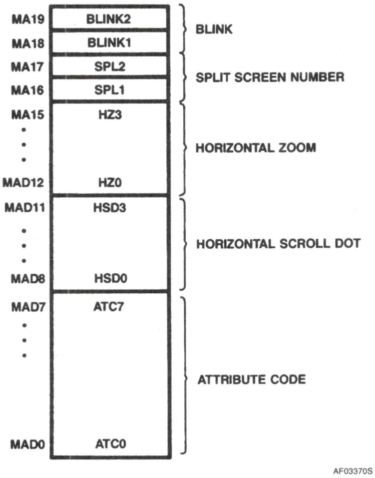


Figure 4. Video Attributes

3. Frame buffer specification in blocks (used for the base register)
4. Back screen color or character color code
5. Display screen selection during screen blink (used with SPL)
6. Interrupt vector address storage
7. Polarity selection of horizontal/vertical synchronization signal, etc.
8. Blinking signal like lamps used in the system
9. Code storage (maximum 8-bit) or selection signal, etc.

Horizontal Scroll (HSD0 – HSD3:MAD8 – MAD11)

These are used in conjunction with external circuitry to implement smooth horizontal scroll. These lines contain the encoded start dot address which is used to control the external shift register load timing and data. HSD usually corresponds to the start dot address of the background screens. However, if the window smooth scroll (SWS) bit of the OMR (operation mode register) is set to 1, HSD outputs the start dot address of the window screen segment. Note that HSD outputs the valid value only within the specified raster area. Changing the register contents during the scanning does not cause any external effects, because the value loaded at the beginning of the area is reserved.

Horizontal Zoom Factor (HZ0 – HZ3:MAD12 – MAD15)

These lines output the encoded (1 – 16) horizontal zoom factor as stored in the zoom factor register (ZFR). Horizontal zoom is accomplished by the ACRTC repeating a single display address and using the HZ outputs to control the external shift register clock. Horizontal zoom can only be applied to the base screen.

Split Position (SPL1 – SPL2:MA16 – MA17)

These lines present the encoded information showing the enabled background screen currently being displayed by the ACRTC.

SPL2	SPL1	
0	0	Background screen not enabled or displayed
0	1	Base screen
1	0	Upper screen
1	1	Lower screen

Even if the split screen display is prohibited, SPL is output if the area is specified.

Blink (BLINK1 – BLINK2:MA18 – MA19)

These lines are used to implement character and screen blink. The lines alternate from high to low periodically as defined in the blink control register (BCR). The blink frequency is specified in units of four field times. A field is defined as the period between successive VSYNC pulses.

ADDRESS SPACE

The ACRTC allows the host to issue commands using logical X-Y coordinate addressing. The ACRTC converts these to physical linear word addresses with bit field offsets in the frame buffer. Figure 5 shows the relationship between a logical X-Y screen address and the frame buffer memory, organized as sequential 16-bit words. The host may specify that a logical pixel consists of 1, 2, 4, 8 or 16 physical bits in the frame buffer. In the example, 4 bits per logical pixel is used allowing 16 colors or tones to be selected.

Up to four logical screens (upper, base, lower and window) are mapped into the ACRTC physical address space. The host specifies a logical physical start address, logical screen physical memory width (number of memory words per raster), logical pixel physical memory width (number of bits per pixel) and the logical origin physical address. Then, logical pixel X-Y addresses issued by the host or by the ACRTC drawing processor are converted to physical frame buffer addresses. The ACRTC also performs bit extraction and masking to map logical pixel operations (in the example, 4 bits) to 16-bit word frame buffer accesses.

The ACRTC has over two hundred bytes of accessible registers. These are organized as hardware-, directly-, and FIFO-accessible (see Figure 6 and Tables 1 and 2).

Hardware-Accessible

The ACRTC is connected to the host MPU as a standard peripheral which occupies two word locations of the host address space. The RS (register select) pin selects one of these two locations. When RS is low, reads access the status register and writes access the address register. The status register summarizes the ACRTC state and is used by the MPU to monitor the overall operation of the ACRTC. The address register is used to program the ACRTC with the address of the specific directly-accessible register which the MPU wishes to access.

Directly-Accessible

These registers are accessed by prior loading of the address register with the chosen register address. Then, when the MPU accesses the ACRTC with RS = 1, the chosen register is accessed. The FIFO entry enables access to FIFO-accessible registers using the ACRTC read and write FIFOs.

The command control register is used to control overall ACRTC operation such as aborting or pausing commands, defining DMA protocols, enabling/disabling interrupt sources, etc. The operation mode register defines basic parameters of ACRTC operation such as frame buffer access mode, display or drawing priority, cursor and display timing skew factors, raster scan mode, etc.

The display control register allows the independent enabling and disabling of each of the four ACRTC logical display screens (base, upper, lower, and window). Also, this register contains the 8 bits of user-defineable video attributes.

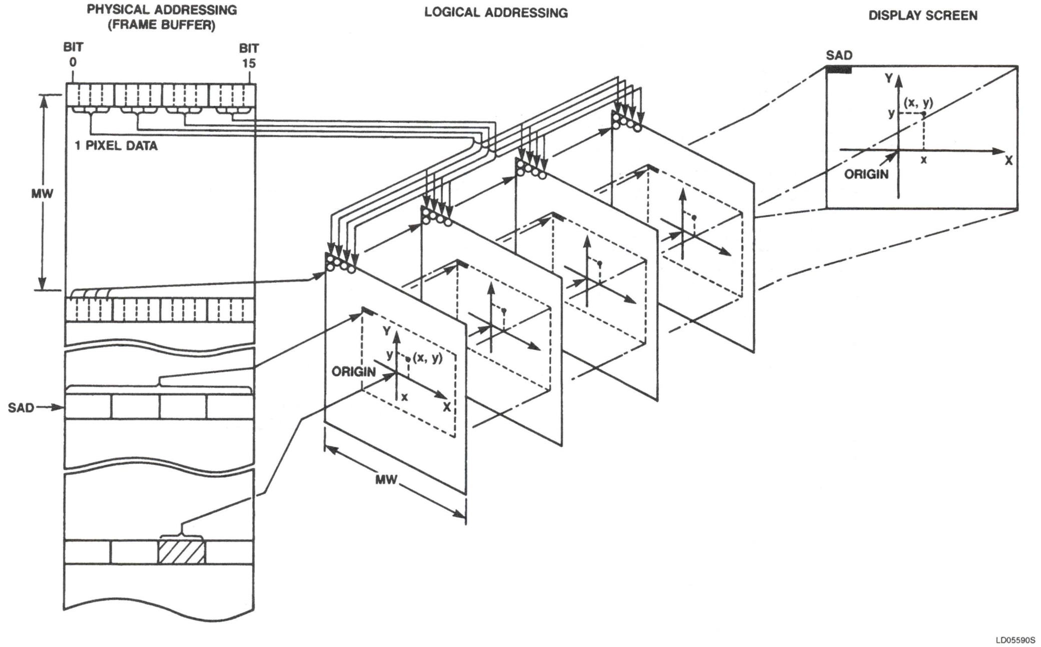


Figure 5. Logical/Physical Addressing

The timing control RAM contains registers which define ACRTC timing. This includes timing specification for CRT control signals (e.g. $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$), logical display screen size and display period, blink timing, etc. The display control RAM contains registers which define logical screen display parameters such as start addresses, raster addresses, and memory width. Also included are the cursor(s) definition, zoom factor, and light pen registers.

FIFO-Accessible

For high-performance drawing, key drawing processor registers are coupled to the host via the ACRTC's separate 16-byte read and write FIFOs. ACRTC commands are sent from the MPU via the write FIFO to the command register. As the ACRTC completes command execution, the next command is automatically fetched from the FIFO into the command register.

The pattern RAM is used to define drawing and painting 'patterns'. The pattern RAM is accessed using the ACRTC's read pattern RAM (RPTN) and write pattern RAM (WPTN) register access commands.

The drawing parameter registers define detailed parameters of the drawing process, such as color control, area control (hitting/clipping) and pattern RAM pointers. The drawing parameter registers are accessed using the ACRTC's read parameter register (RPR) and write parameter register (WPR) register access commands.

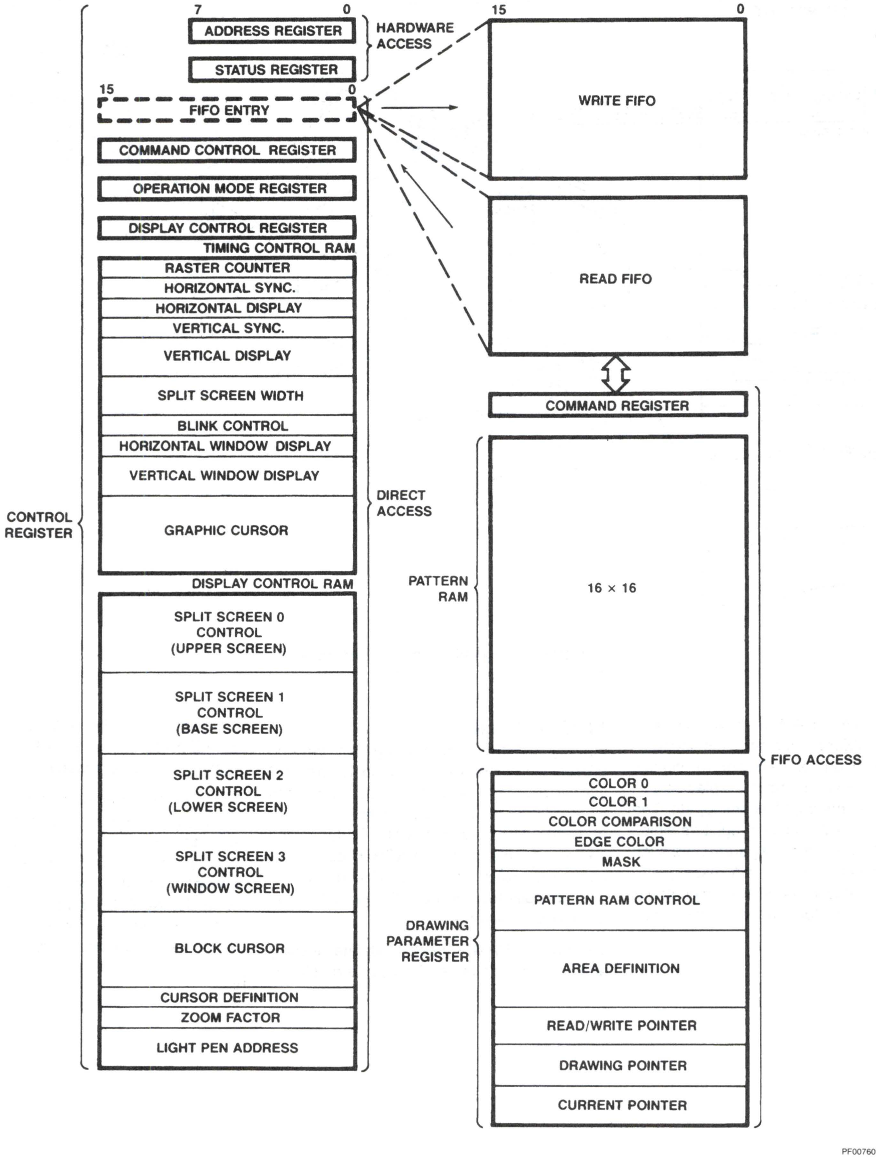

CONTROL REGISTER
7
0
ADDRESS REGISTER
STATUS REGISTER
HARDWARE ACCESS
15
0
FIFO ENTRY
COMMAND CONTROL REGISTER
OPERATION MODE REGISTER
DISPLAY CONTROL REGISTER
TIMING CONTROL RAM
RASTER COUNTER
HORIZONTAL SYNC.
HORIZONTAL DISPLAY
VERTICAL SYNC.
VERTICAL DISPLAY
SPLIT SCREEN WIDTH
BLINK CONTROL
HORIZONTAL WINDOW DISPLAY
VERTICAL WINDOW DISPLAY
GRAPHIC CURSOR
DIRECT ACCESS
DISPLAY CONTROL RAM
SPLIT SCREEN 0 CONTROL (UPPER SCREEN)
SPLIT SCREEN 1 CONTROL (BASE SCREEN)
SPLIT SCREEN 2 CONTROL (LOWER SCREEN)
SPLIT SCREEN 3 CONTROL (WINDOW SCREEN)
BLOCK CURSOR
CURSOR DEFINITION
ZOOM FACTOR
LIGHT PEN ADDRESS
15
0
WRITE FIFO
READ FIFO
COMMAND REGISTER
PATTERN RAM
16 × 16
FIFO ACCESS
COLOR 0
COLOR 1
COLOR COMPARISON
EDGE COLOR
MASK
PATTERN RAM CONTROL
AREA DEFINITION
READ/WRITE POINTER
DRAWING POINTER
CURRENT POINTER
DRAWING PARAMETER REGISTER
PF00760S

Figure 6. Programming Model

Table 1. Programming Model (Hardware Access, Direct Access Registers)

CS	RS	RW	REG. NO.	REGISTER NAME		ABBRE.	DATA (H)								DATA (L)							
							15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	—	—	—	—		—	....								....							
0	0	0	AR	Address Register		AR	....								Address							
0	0	1	SR	Status Register		SR	....								CER	ARD	CED	LPD	RFF	RFR	WFR	WFE
		1/0	r00	FIFO Entry		FE	FE															
		1/0	r02	Command Control		CCR	ABT	PSE	DDM	CDM	DRC	GBM			CRE	ARE	CEE	LPE	RFE	RRE	WRE	WEE
		1/0	r04	Operation Mode		OMR	M/S	STR	ACP	WSS	CSK		DSK		RAM	GAI			ACM		RSM	
		1/0	r06	Display Control		DCR	DSP	SE1	SE0		SE2		SE3		ATR							
		—	r08 ↓ r7E	(undefined)		—	....															
		1	r80	Raster Count		RCR	....				RC											
		1/0	r82	Horizontal Sync.		HSR	HC								....			HSW				
			r84	Horizontal Display		HDR	HDS								HDW							
		1/0	r86	Vertical Sync.		VSR	....				VC											
		1/0	r88	Vertical Display		VDR	VDS								....			VSW				
		1/0	r8A	Split Screen Width		SSW	....				SP1											
		1/0	r8C				....				SP0											
		1/0	r8E				....				SP2											
		1/0	r90	Blink Control		BCR	BON1				BOFF1				BON2				BOFF2			
		1/0	r92	Horizontal Window Display		HWR	HWS								HWW							
		1/0	r94	Vertical Window Display		VWR	....				VWS											
		1/0	r96				....				VWW											
		1/0	r98	Graphic Cursor		GCR	CXE								CXS							
		1/0	r9A				....				CYS											
		1/0	r9C				....				CYE											
		—	r9E	(undefined)		—	....															
		—	rA0 ↓ rBE																			
0	1	1/0	rC0	Upper Screen	Raster Addr. 0	RAR0	....			LRA0					....			FRA0				
		1/0	rC2		Memory Width 0	MWR0	CHR	....			MW0											
		1/0	rC4		Start Addr. 0	SAR0	....				SDA0				....			SA0H/SRA0				
		1/0	rC6				SA0L															
		1/0	rC8	Base Screen	Raster Addr. 1	RAR1	....			LRA1					....			FRA1				
		1/0	rCA		Memory Width 1	MWR1	CHR	....			MW1											
		1/0	rCC		Start Addr. 1	SAR1	....				SDA1				....			SA1H/SRA1				
		1/0	rCE				SA1L															
		1/0	rD0	Upper Screen	Raster Addr. 2	RAR2	....			LRA2					....			FRA2				
		1/0	rD2		Memory Width 2	MWR2	CHR	....			MW2											
		1/0	rD4		Start Addr. 2	SAR2	....				SDA2				....			SA2H/SRA2				
		1/0	rD6				SA2L															
		1/0	rD8	Window Screen	Raster Addr. 3	RAR3	....			LRA3					....			FRA3				
		1/0	rDA		Memory Width 3	MWR3	CHR	....			MW3											
		1/0	rDC		Start Addr. 3	SAR3	....				SDA3				....			SA3H/SRA3				
		1/0	rDE				SA3L															
		1/0	rE0	Block Cursor 1		BCUR1	BCW1			BCSR1					....			BCER1				
		1/0	rE2				BCA1															
		1/0	rE4	Block Cursor 2		BCUR2	BCW2			BCSR2					....			BCER2				
		1/0	rE6				BCA2															
		1/0	rE8	Cursor Definition		CDR	CM		CON1			COFF1			....		CON2			COFF2		
		1/0	rEA	Zoom Factor		ZFR	HZF				VZF				....							
		1	rEC	Light Pen Address		LPAR	....								CHR	....			LPAH			
		1	rEE				LPAL															
		—	rF0 ↓ rFE	(undefined)		—	....															

NOTE: { 1 .... "High" level
0 .... "Low" level

HARDWARE ACCESS, DIRECT ACCESS REGISTERS

ABT: Abort
ACM: Access Mode
ACP: Access Priority
Address: Register No. of the control register
ARD: Area Detect
ARE: Area Detect Interrupt Enable
ATR: Attribute Control
CDM: Command DMA Mode
CED: Command End
CEE: Command End Interrupt Enable
CER: Command Error
CRE: Command Error Interrupt Enable
CSK: Cursor Display Skew
DDM: Data DMA Mode
DRC: DMA Request Control
DSK: $\overline{\text{DISP}}$ Skew
DSP: $\overline{\text{DISP}}$ Signal Control
FE: FIFO Entry
GAI: Graphic Address Increment Mode
GBM: Graphic Bit Mode
HC: Horizontal Cycle
HDS: Horizontal Display Start
HDW: Horizontal Display Width
HSW: Horizontal Sync. Width
LPD: Light Pen Strobe Detect
LPE: Light Pen Strobe Interrupt Enable
M/S: Master/Slave
PSE: Pause
RAM: RAM Mode
RC: Raster Count
RFE: Read FIFO Full Interrupt Enable
RFF: Read FIFO Full
RFR: Read FIFO Ready
RRE: Read FIFO Ready Interrupt Enable
RSM: Raster Scan Mode
SE0: Split Enable 0
SE1: Split Enable 1
SE2: Split Enable 2
SE3: Split Enable 3
STR: Start
VC: Vertical Cycle
VDS: Vertical Display Start
VSW: Vertical Sync. Width
WEE: Write FIFO Empty Interrupt Enable
WFE: Write FIFO Empty
WFR: Write FIFO Ready
WRE: Write FIFO Ready Interrupt Enable
WSS: Window Smooth Scroll
SP0, SP1, SP2: Split Screen 0 Width, Split Screen 1 Width, Split Screen 2 Width
BON1, BON2: Blink ON 1, Blink ON 2
BOFF1, BOFF2: Blink OFF 1, Blink OFF 2
HWS: Horizontal Window Start
HWW: Horizontal Window Width
VWS: Vertical Window Start
VWW: Vertical Window Width
CXS, CYS: Cursor X Start, Cursor Y Start
CXE, CYE: Cursor X End, Cursor Y End
FRA: First Raster Address
LRA: Last Raster Address
CHR: Character
MW: Memory Width
SDA: Start Dot Address
SAH/SRA: Start Address "High"/Start Raster Address
SAL: Start Address "Low"
BCW1, BCW2: Block Cursor Width 1, Block Cursor Width 2
BCSR1, BCSR2: Block Cursor Start Raster 1, Block Cursor Start Raster 2
BCER1, BCER2: Block Cursor End Raster 1, Block Cursor End Raster 2
BCA1, BCA2: Block Cursor Address 1, Block Cursor Address 2
CM: Cursor Mode
CON1, CON2: Cursor ON 1, Cursor ON 2
COFF1, COFF2: Cursor OFF 1, Cursor OFF 2
HZF, VZF: Horizontal Zoom Factor, Vertical Zoom Factor
LPAH: Light Pen Address "High"
LPAL: Light Pen Address "Low"

Table 2. Programming Model (Drawing Parameter Registers)

REGISTER NO.	READ/ WRITE	NAME OF REGISTER	ABBR.	DATA (H)								DATA (L)							
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pr00	R/W	Color 0	CL0	CL0															
Pr01	R/W	Color 1	CL1	CL1															
Pr02	R/W	Color Comparison	CCMP	CCMP															
Pr03	R/W	Edge Color	EDG	EDG															
Pr04	R/W	Mask	MASK	MASK															
Pr05	R/W	Pattern RAM Control	PRC	PPY				PZCY				PPX				PZCX			
↓				PSY				▨				PSX				▨			
Pr07				PEY				PZY				PEX				PZX			
Pr08	R/W	Area Definition **	ADR	XMIN															
↓				YMIN															
				XMAX															
Pr0B				YMAX															
Pr0C Pr0D	R/W	Read/Write Pointer	RWP	DN		▨						RWPH							
				RWPL												▨			
Pr0E Pr0F	—	....	—	—															
				—															
Pr10 Pr11	R	Drawing Pointer	DP	DN		▨						DPAH							
				DPAL												DPD			
Pr12 Pr13	R	Current Pointer **	CP	X															
				Y															
Pr14 Pr15	—	....	—	—															
				—															

NOTES:

▨ .... Always set to "0"

** .... Set binary complements for negative values of X and Y axis.

DRAWING PARAMETER REGISTER

R: Register which can be read by Read Parameter Register command (RPR)
W: Register which can be written into by Write Parameter Register command (WPR)
—: Access is not allowed
CL0: Defines the color data used for the drawing when logical drawing data = 0
CL1: Defines the color data used for the drawing when logical drawing data = 1
CCMP: Defines the comparative color of the drawing operation
EDG: Defines the edge color
MASK: Defines the bit pattern used to mask bits upon which data transfer should not be performed
PSX, PSY: Pattern Start Point
PEX, PEY: Pattern End Point
PPX, PPY: Pattern Scan Start Point
PZX, PZY: Pattern Zoom
PZCX, PZCY: Pattern Zoom Count
XMIN, YMIN: Start point of Area definition
XMAX, YMAX: End point of Area definition
DN: Screen Number
RWPH: High-order 8 bits of Read Write Pointer Address
RWPL: Low-order 12 bits of Read Write Pointer Address
DPAH: High-order 8 bits of Drawing Pointer Address
DPAL: Low-order 12 bits of Drawing Pointer Address
DPD: Drawing Pointer Dot Address
X, Y: Position indicated by Current Pointer on X-Y coordinate

COMMANDS

The ACRTC has 38 commands classified into three groups — register access, data transfer, and graphic drawing (see Table 3). Five register access commands allow access to drawing processor, drawing parameter registers, and the pattern RAM. Ten data transfer commands are used to move data between the host system memory and the frame buffer, or within the frame buffer. Twenty-three graphic drawing commands cause the ACRTC to perform drawing operations. Parameters for these commands are specified using logical X-Y addressing.

All of the above commands, parameters and data are transferred via the ACRTC read and write FIFOs.

Assuming the ACRTC has been properly initialized, the MPU must perform two steps to cause graphic drawing. First, the MPU must specify certain drawing parameters which define a number of details associated with the drawing process. For example, to draw a figure or paint an area, the MPU must specify the drawing or painting 'pattern' by initializing the ACRTC pattern RAM and related pointers. Also, if clipping and hitting control are desired, the MPU specifies the 'area' to be monitored during drawing by initializing area definition registers. Other drawing parameters include color, edge definition, etc.

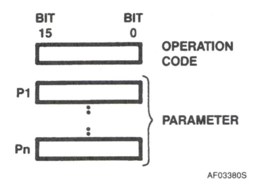


Figure 7. 16-Bit Interface

After the drawing parameters have been specified, the MPU issues a graphic drawing command and any required command parameters, such as the CRCL (circle) command with a radius parameter. The ACRTC then performs the specified drawing operation by reading, modifying and rewriting the contents of the frame buffer.

Command Format

ACRTC commands consist of a 16-bit opcode, optionally followed by one or more 16-bit parameters. When 8-bit MPU mode is used, commands, parameters, and data are sent to and from the ACRTC in the order of high-byte, low-byte.

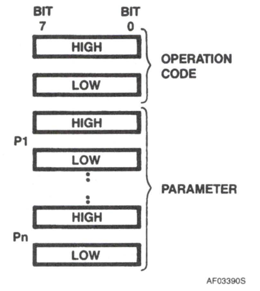


Figure 8. 8-Bit Interface

16-Bit Interface — In the case of 16-bit interface, first move the 16-bit operation code and then move necessary 16-bit parameters one by one (see Figure 7).

8-Bit Interface — In the case of 8-bit interface, first move the operation code's high byte followed by low byte, and then move the parameters in the same order (see Figure 8).

REGISTER ACCESS COMMAND

MNEMONIC	OPERATION CODE					PARAMETER	NO. WORDS	~ (CYCLES)
ORG	0 0 0 0	0 1	0 0	0 0 0 0	0 0 0 0	DPH DPL	3	8
WPR	0 0 0 0	1 0	0 0	0 0 0	RN	D	2	6
RPR	0 0 0 0	1 1	0 0	0 0 0	RN		1	6
WPTN	0 0 0 1	1 0	0 0	0 0 0 0	PRA	n D_1,................,D_n	n + 2	4n + 8
RPTN	0 0 0 1	1 1	0 0	0 0 0 0	PRA	n	2	4n + 10

RN : Register number of the drawing parameter register ($0 – $13)
PRA: Pattern RAM address at which Read/Write operation starts ($0 – $F)
DPH: Drawing pointer register High word
DPL: Drawing pointer register Low word

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPH	DN		0	0	0	0	0	0	DPAH							
DPL	DPAL												DPD			

DPAH: Higher 8 bits of Drawing Pointer address
DPAL: Lower 12 bits of Drawing Pointer address
DPD: Dot position in the memory address
D, D_1,........,D_n: Write data
n: Number of Read/Write data

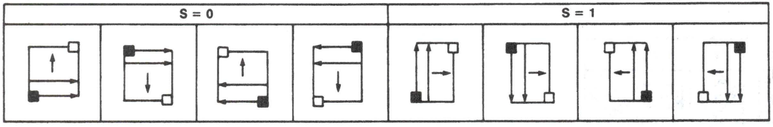
DN	SCREEN NO.
00	Upper Screen
01	Base Screen
10	Lower Screen
11	Window Screen

DATA TRANSFER COMMAND

MM: Modify Mode

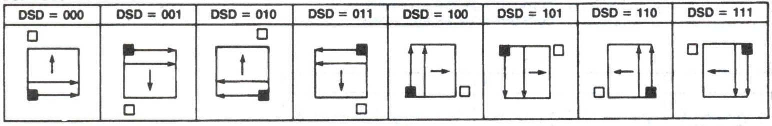
MM	FUNCTION	
00	Replace	Replace drawing point data with modifier information
01	OR	OR drawing point data with modifier data and rewrite the result data to the frame buffer
10	AND	AND drawing point data with modifier data and rewrite the result data to the frame buffer
11	Ex-OR	Ex-OR drawing point data with modifier data and rewrite the result data to the frame buffer

S : SOURCE SCAN DIRECTION



■ : SOURCE AREA START ADDRESS
☐ : SOURCE AREA END ADDRESS

DSD : DESTINATION SCAN DIRECTION



■ : READ WRITE POINTER START POINT
☐ : READ WRITE POINTER END POINT

DF05580S

AX: Number of word in X-axis direction-1
AY: Number of word in Y-axis direction-1
D: Write data
SAH: Source Start Address High word
SAL: Source Start Address Low word

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAH	0	0	0	0	0	0	0	0	(SAH)							
SAL	(SAL)												0	0	0	0

(SAH): Memory address Higher 8 bits
(SAL): Memory address Lower 12 bits
x: Number of word in X-axis direction
y: Number of word in Y-axis direction
↑: Rounding up

GRAPHIC DRAWING COMMAND

AREA : Area Mode
COL : Color Mode
OPM : Operation Mode

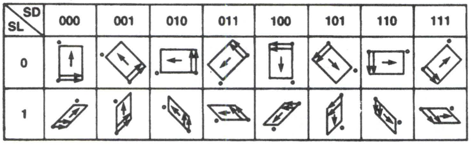
C: Circling Direction

C	DIRECTION
0	Counterclockwise
1	Clockwise

E: Definition of Edge color

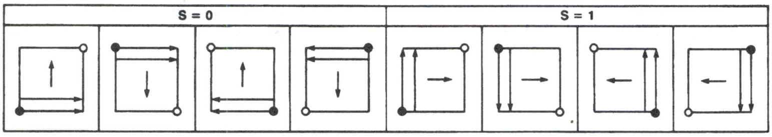
E	DEFINITION
0	Edge color is defined by the data in the edge color register.
1	Edge color is defined by the data excluding the above.

SL : SLANT, SD : SCAN DIRECTION



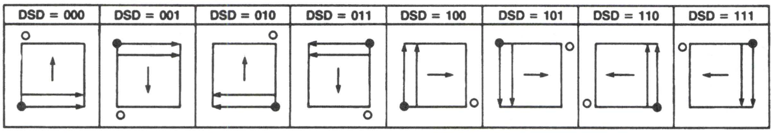
● : CURRENT POINTER START POINT
○ : CURRENT POINTER END POINT

S : SOURCE SCAN DIRECTION



● : SOURCE AREA START DOT POSITION
○ : SOURCE AREA END DOT POSITION

DSD : DESTINATION SCAN DIRECTION



● : CURRENT POINTER START POINT
○ : CURRENT POINTER END POINT

DF05690S

X, X1, . . ., X_n : Absolute X-address from the original point
Y, Y1, . . ., Y_n : Absolute Y-address from the original point
dX : Relative X-address from the current pointer
dY : Relative Y-address from the current pointer
n : Number of notes
dX1, . . ., dXn : Relative X-address from each node
dY1, . . ., dYn : Relative Y-address from each node
r : Dot number on radius
a, b : $(DX)^2$: $(DY)^2 = a : b$
DX : X-direction dot number
DY : Y-direction dot number
Xc : Absolute X-address of the center point of arc/ellipse
Yc : Absolute Y-address of the center point of arc/ellipse
dXc : Relative X-address from the current pointer to the center point of arc/ellipse
dYc : Relative Y-address from the current pointer to the center point of arc/ellipse
Xe : Absolute X-address of the end point of arc/ellipse
Ye : Absolute Y-address of the end point of arc/ellipse
dXe : Relative X-address from the current pointer to the end point of arc/ellipse
dYe : Relative Y-address from the current pointer to the end point of arc/ellipse
Xs : Absolute X-address of the start dot position in source area
Ys : Absolute Y-address of the start dot position in source area
dXs : Relative X-address from the current pointer to the state dot position in source area
dYs : Relative Y-address from the current pointer to the state dot position in source area
P : 4(OPM = 000 – 011)/6(OPM = 100 – 111)
L, L0 : Dot number on straight line
d : Total dot number
A : Scan main direction dot number
B : Scan sub direction dot number

Table 3. ACRTC Command Table

TYPE	MNEMONIC	COMMAND NAME	OPERATION CODE	PARAMETER	NO. WORDS	(CYCLES)[3]
Register Access Command	ORG	Origin	0 0 0 0 ¦ 0 1 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 0 0	DPH DPL	3	8
	WPR	Write Parameter Register	0 0 0 0 ¦ 1 0 ¦ 0 0 ¦ 0 0 0 ¦ RN	D	2	6
	RPR	Read Parameter Register	0 0 0 0 ¦ 1 1 ¦ 0 0 ¦ 0 0 0 ¦ RN		1	6
	WPTN	Write Pattern RAM	0 0 0 1 ¦ 1 0 ¦ 0 0 ¦ 0 0 0 0 ¦ PRA	n $D_1,\ldots,D_n$	n + 2	4n + 8
	RPTN	Read Pattern RAM	0 0 0 1 ¦ 1 1 ¦ 0 0 ¦ 0 0 0 0 ¦ PRA	n	2	4n + 10
Data Transfer Command	DRD	DMA Read	0 0 1 0 ¦ 0 1 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 0 0	AX AY	3	(4x + 8)y + 12[x • y/8↑] + (62 ~ 68)
	DWT	DMA Write	0 0 1 0 ¦ 1 0 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 0 0	AX AY	3	(4x + 8)y + 16[x • y/8↑] + 34
	DMOD	DMA Modify	0 0 1 0 ¦ 1 1 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 ¦ MM	AX AY	3	(4x + 8)y + 16[x • y/8↑] + 34
	RD	Read	0 1 0 0 ¦ 0 1 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 0 0		1	
	WT	Write	0 1 0 0 ¦ 1 0 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 0 0	D	2	8
	MOD	Modify	0 1 0 0 ¦ 1 1 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 ¦ MM	D	2	8
	CLR	Clear	0 1 0 1 ¦ 1 0 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 0 0	D AX AY	4	(2x + 8)y + 12
	SCLR	Selective Clear	0 1 0 1 ¦ 1 1 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 ¦ MM	D AX AY	4	(4x + 6)y + 12
	CPY	Copy	0 1 1 0 ¦ S ¦ DSD ¦ 0 0 0 0 ¦ 0 0 0 0	SAH SAL AX AY	5	(6x + 10)y + 12
	SCPY	Selective Copy	0 1 1 1 ¦ S ¦ DSD ¦ 0 0 0 0 ¦ 0 0 ¦ MM	SAH SAL AX AY	5	(6x + 10)y + 12
Graphic Command	AMOVE	Absolute Move	1 0 0 0 ¦ 0 0 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 0 0	X Y	3	56
	RMOVE	Relative Move	1 0 0 0 ¦ 0 1 ¦ 0 0 ¦ 0 0 0 0 ¦ 0 0 0 0	dX dY	3	56
	ALINE	Absolute Line	1 0 0 0 ¦ 1 0 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	X Y	3	P • L + 18
	RLINE	Relative Line	1 0 0 0 ¦ 1 1 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	dX dY	3	P • L + 18
	ARCT	Absolute Rectangle	1 0 0 1 ¦ 0 0 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	X Y	3	2P(A + B) + 54
	RRCT	Relative Rectangle	1 0 0 1 ¦ 0 1 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	dX dY	3	2P(A + B) + 54
	APLL	Absolute Polyline	1 0 0 1 ¦ 1 0 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	n X1, Y1,...Xn, Yn	2n + 2	Σ [P • L + 16] + 8
	RPLL	Relative Polyline	1 0 0 1 ¦ 1 1 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	n dX1, dY1,...dXn, dYn	2n + 2	Σ [P • L + 16] + 8
	APLG	Absolute Polygon	1 0 1 0 ¦ 0 0 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	n X1, Y1,...Xn, Yn	2n + 2	Σ [P • L + 16] + P • Lo + 20
	RPLC	Relative Polygon	1 0 1 0 ¦ 0 1 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	n dX1, dY1,...dXn, dYn	2n + 2	Σ [P • L + 16] + P • Lo + 20
	CRCL	Circle	1 0 1 0 ¦ 1 0 ¦ 0 ¦ C ¦ AREA ¦ COL ¦ OPM	r	2	8d + 66
	ELPS	Ellipse	1 0 1 0 ¦ 1 1 ¦ 0 ¦ C ¦ AREA ¦ COL ¦ OPM	a b dX	4	10d + 90
	AARC	Absolute Arc	1 0 1 1 ¦ 0 0 ¦ 0 ¦ C ¦ AREA ¦ COL ¦ OPM	Xc Yc Xe Ye	5	8d + 18
	RARC	Relative Arc	1 0 1 1 ¦ 0 1 ¦ 0 ¦ C ¦ AREA ¦ COL ¦ OPM	dXc dYc dXe dYe	5	8d + 18
	AEARC	Absolute Ellipse Arc	1 0 1 1 ¦ 1 0 ¦ 0 ¦ C ¦ AREA ¦ COL ¦ OPM	a b Xc Yc Xe Ye	7	10d + 96
	REARC	Relative Ellipse Arc	1 0 1 1 ¦ 1 1 ¦ 0 ¦ C ¦ AREA ¦ COL ¦ OPM	a b dXc dYc dXe dYe	7	10d + 96
	AFRCT	Absolute Filled Rectangle	1 1 0 0 ¦ 0 0 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	X Y	3	(P • A + B)B + 18
	RFRCT	Relative Filled Rectangle	1 1 0 0 ¦ 0 1 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM	dX dY	3	(P • A + B)B + 18
	PAINT	Paint	1 1 0 0 ¦ 1 0 ¦ 0 ¦ E ¦ AREA ¦ 0 0 ¦ 0 0 0		1	(18A + 102)B–58[1]
	DOT	Dot	1 1 0 0 ¦ 1 1 ¦ 0 0 ¦ AREA ¦ COL ¦ OPM		1	8
	PTN	Pattern	1 1 0 1 ¦ SL ¦ SD ¦ AREA ¦ COL ¦ OPM	SZ[2]	2	(P • A + 10)B + 20
	AGCPY	Absolute Graphic Copy	1 1 1 0 ¦ S ¦ DSD ¦ AREA ¦ 0 0 ¦ OPM	Xs Ys DX DY	5	((P + 2)A + 10)B + 70
	RGCPY	Relative Graphic Copy	1 1 1 1 ¦ S ¦ DSD ¦ AREA ¦ 0 0 ¦ OPM	dXs dYs DX DY	5	((P + 2)A + 10)B + 70

NOTES:

1. In case of rectangular filling

2. SZ:

15 8	7 0
SZy	SZx

SZy, SZx: Pattern Size

n: number of repetition X/Y: drawing words of X-direction/Y-direction
L/Lo/d: sum of drawing dots A/B: drawing dots of main/sub direction
E: [E = 0 (Stop at Edge color), E = 1 (Stop at excepting Edge color)] C: [C = 1 (clockwise), C = 0 (reverse)]
[↑]: rounding up
P = 4: OPM-000 – 011
6: OPM-100 – 111

3. Cycles: 2 clock cycle time

PROGRAM TRANSFER

Program transfer occurs when the MPU specifies the FIFO entry address and then writes commands/parameters to the write FIFO under program control (RS = high; R/$\overline{W}$, $\overline{CS}$ = low). The MPU writes are normally synchronized with the ACRTC FIFO status by software polling or interrupts.

Software Polling (WFR, WFE Interrupts Disabled)

1. MPU program checks the SR (status register) for write FIFO ready (WFR) flag = 1, and then writes 1-word op-code/parameters.
2. MPU program checks the SR (status register) for write FIFO empty (WFE) flag = 1, and then writes 1- to 8-word op-code/parameters.

Interrupt-Driven (WFR, WFE Interrupts Enabled)

1. MPU WFR interrupt service routine writes 1-word op-code/parameters.
2. MPU WFE interrupt service routine writes 1- to 8-word op-code/parameters.

In the specific case of register access commands and an initially empty write FIFO, MPU writes need not be synchronized to the write FIFO status. The ACRTC can fetch and execute these commands faster than the MPU can issue them.

COMMAND DMA TRANSFER

Commands and parameters can be transferred from MPU system memory using an external DMAC. The MPU initiates and terminates command DMA transfer mode under software control (CDM bit of CCR). Command DMA can also be terminated by assertion of the ACRTC $\overline{DONE}$ signal. $\overline{DONE}$ is treated as an input in command DMA transfer mode.

Using command DMA transfer, the ACRTC will issue cycle stealing DMA requests to the DMAC when the write FIFO is empty. The DMA data is automatically sent from system memory to the ACRTC write FIFO regardless of the contents of the address register.

NOTES:

1. Ensure that the write FIFO is empty and all the commands are terminated before starting the command DMA transfer.
2. The data DMA command cannot be executed in the command DMA transfer mode.

Register Access Commands

Registers associated with the drawing processor (pattern RAM and drawing parameter registers) are accessed through the read and write FIFOs using the register access commands.

Data Transfer Commands

Data transfer commands are used to move blocks of data between the MPU system memory and the ACRTC frame buffer or within the frame buffer itself. Before issuing these commands, a physical 20-bit frame buffer address must be specified in the RWP (read/write pointer) drawing parameter register.

Tables 4 and 5 list register access commands and data transfer commands. Figure 9 shows the data transfer command format.

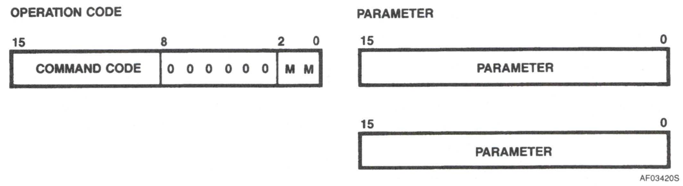


Figure 9. Data Transfer Command Format

Table 4. Register Access Commands

COMMAND	FUNCTION
ORG	Initialize the relation between the origin point in the X-Y coordinates and the physical address
WPR	Write into the parameter register
RPR	Read the parameter register
WPTN	Write into the pattern RAM
RPTN	Read the pattern RAM

Table 5. Data Transfer Commands

COMMAND	FUNCTION
DRD	Transfer data, by DMA transfer, from the frame buffer to the MPU system memory
DWT	Transfer data, by DMA transfer, from the MPU system memory to the frame buffer
DMOD	Transfer data, by DMA transfer, from the MPU system to the frame buffer subject to logical modification (bit-maskable)
RD	Read one word of data from the frame buffer specified by the read/write pointer (RWP), and load the word into Read FIFO
WT	Write one word of data to the frame buffer specified by the read/write pointer (RWP)
MOD	Perform logical operation on one word in the frame buffer specified by the read/write pointer (RWP) (bit-maskable)
CLR	Clear a rectangular area of the frame buffer with a data in the command parameter
SCLR	Initialize a rectangular area of the frame buffer with 1-word data subject to logical operation (bit-maskable)
CPY	Copy frame buffer data from one area (source area) to another area (destination area) specified by the read/write pointer (RWP).
SCPY	Copy frame buffer data from one area (source area) to another area (destination area) subject to logical modification by word. The source and destination areas must reside on the same screen (bit-maskable)

MODIFY MODE

The DMOD, MOD, SCLR and SCPY commands allow four types of bit-level logical operations to be applied to frame buffer data. The modify mode is encoded in the lower two bits (MM) of these op-codes. The bit positions within each frame buffer word to be modified are selectable using the mask register (MASK). Bits set to 1 are modifiable, ones to 0 are masked and not modifiable.

MM		Modify Mode
0	0	REPLACE frame buffer data with command parameter data
0	1	OR frame buffer data with command parameter data and rewrite to the frame buffer
1	0	AND frame buffer data with command parameter data and rewrite to the frame buffer
1	1	Ex-OR frame buffer data with command parameter data and rewrite to the frame buffer

GRAPHIC DRAWING COMMANDS

The ACRTC has 23 separate graphic drawing commands (see Table 6). Graphic drawing is performed by modifying the contents of the frame buffer based on microcoded drawing algorithms in the ACRTC drawing processor.

Most coordinate parameters for graphic drawing commands are specified using logical pixel X-Y addressing.

The complex task of translating a logical pixel address to a linear frame buffer word address, and further selecting the appropriate sub-field of the word is performed at high speed by ACRTC hardware. For example, a given logical pixel in 4 bits per logical pixel mode might reside in bits 8 – 11 of a frame buffer word.

Many instructions allow specification of X-Y coordinates with either absolute or relative X-Y coordinates (e.g., ALINE and RLINE). In both cases, two's complement numbers are used to represent positive and negative result values.

Absolute Coordinate Specification

The screen address (X, Y) is specified in units of logical pixels relative to an origin point defined with the ORG command (see Figure 10).

Relative Coordinate Specification

The screen address (dX, dY) is specified in units of logical pixels relative to the current drawing pointer (CP) position. A graphic drawing command consists of a 16-bit op-code and optionally 0 to 64k 16-bit parameters (see Figure 11). The 16-bit op-code consists of an 8-bit command code, an area mode specifier (3 bits), a color mode specifier (2 bits) and an operation mode specifier (3 bits).

Table 6. Graphic Drawing Commands

COMMAND	FUNCTION
AMOVE	Move the current pointer (CP) to an absolute logical pixel X-Y address
RMOVE	Move the CP to a relative logical pixel X-Y address
ALINE	Draw a straight line from the CP to a command-specified endpoint of the absolute coordinates
RLINE	Draw a straight line from the CP to a command-specified endpoint of the relative coordinates
ARCT	Draw a rectangle defined by the CP and a command-specified diagonal point of the absolute coordinates
RRCT	Draw a rectangle defined by the CP and a command-specified diagonal point of the relative coordinates
APLL	Draw a polyline (multiple contiguous segments) from the CP through command-specified points of the absolute coordinates
RPLL	Draw a polyline (multiple contiguous segments) from the CP through command-specified points of the relative coordinates
APLG	Draw a polygon which connects the start pointer (CP) and command-specified points of the absolute coordinates
RPLG	Draw a polygon which connects the CP and command-specified points of the relative coordinates
CRCL	Draw a circle of the radius, R, placing the CP at the center
ELPS	Draw an ellipse whose shape is specified by command parameters, placing the CP at the center
AARC	Draw an arc by using the CP as a start point with an end point and a center point of the absolute coordinates
RARC	Draw an arc by using the CP as a start point with an end point and a center point of the relative coordinates.
AEARC	Draw an ellipse arc by using the CP as a start point with an end point and a center point of the absolute coordinates
REARC	Draw an ellipse arc by using the CP as a start point with an end point and a center point of the relative coordinates
AFRCT	Paint a rectangular area specified by the CP and command parameters (absolute coordinates) according to a figure pattern stored in the pattern RAM (tiling).
RFRCT	Paint a rectangular area specified by the CP and command parameters (relative coordinates) according to a figure pattern stored in the pattern RAM (tiling).
PAINT	Paint a closed area surrounded by edge color using a figure pattern stored in the pattern RAM (tiling).
DOT	Mark a dot on the coordinates where the CP indicates
PTN	Draw a graphic pattern defined in the pattern RAM onto a rectangular area specified by the CP and by the pattern size (rotation angle: 45°)
AGCPY	Copy a rectangular area specified by the absolute coordinates to the address specified by the CP, (rotation angle: 90°/mirror turnover)
RGCPY	Copy a rectangular area specified by the relative coordinates to the address specified by the CP (rotation angle: 90°/mirror turnover)

The area mode allows versatile clipping and hitting detection. A drawing area can be defined, and should drawing operations attempt to enter or leave that area, a number of programmable actions can be taken by the ACRTC.

Absolute Coordinate Specification
Specifies the addresses (x, y) based on the origin point set by the ORG command.

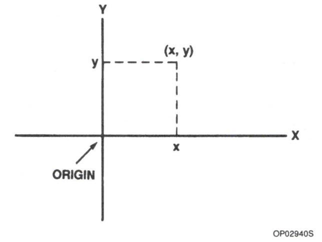


Figure 10. Absolute Coordinate Specification

Relative Coordinate Specification
Specifies the relative addresses (Δx, Δy) related to the current drawing point.

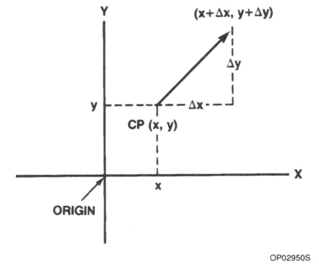


Figure 11. Relative Coordinate Specification

The color mode determines whether the pattern RAM is used indirectly to select color registers or is directly used as the color information.

The operation mode defines one of eight logical operations to be performed between the frame buffer read data and the color data in the pattern RAM to determine the drawing data to be rewritten into the frame buffer. Figure 12 shows the graphic drawing command format.

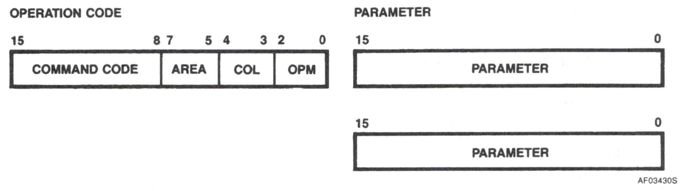


Figure 12. Graphic Drawing Command Format

OPERATION MODE

The operation mode (OPM bits) of the graphic drawing command specify the logical drawing condition.

Figure 13 shows examples of a drawing pattern applied with various OPM modes.

OPM	Operation Mode
0 0 0	REPLACE: * Replaces the frame buffer data with the color data
0 0 1	OR: * ORs the frame buffer data with the color data. The result is rewritten to the frame buffer
0 1 0	AND: * ANDs the frame buffer data with the color data. The result is rewritten to the frame buffer
0 1 1	Ex-OR: * Ex-ORs the frame buffer data with the color data. The result is rewritten to the frame buffer.
1 0 0	CONDITIONAL REPLACE (read data = CCMP): * When the frame buffer data at the drawing position is equal to the comparison color (CCMP), the frame buffer data is replaced with the color data
1 0 1	CONDITIONAL REPLACE (read data ≠ CCMP): * When the frame buffer data at the drawing position is not equal to the comparison color (CCMP), the frame buffer data is replaced with the color data
1 1 0	CONDITIONAL REPLACE (read data < CL): * ** When the frame buffer data at the drawing position is less than the color register data (CL), the frame buffer data is replaced with the color data
1 1 1	CONDITIONAL REPLACE (read data > CL): * ** When the frame buffer data at the drawing position is greater than the color register data (CL), the frame buffer data is replaced with the color data.

COLOR MODE

The color mode (COL bits) specify the source of the drawing color data as directly or indirectly (using the color registers) determined by the contents of the pattern RAM.

COL	Color Mode
0 0	When pattern RAM data = 0, color register 0 is used When pattern RAM data = 1, color register 1 is used
0 1	When pattern RAM data = 0, drawing is suppressed When pattern RAM data = 1, color register 1 is used
1 0	When pattern RAM data = 0, color register 0 is used When pattern RAM data = 1, drawing is suppressed
1 1	Pattern RAM contents are directly used as color data

The color mode chooses the source for color information based on the contents (0 or 1) of a particular bit in the 16-bit by 16-bit (32-byte) pattern RAM. A sub-pattern is specified by programming the pattern RAM control register (PRC) with the start (PSX, PSY) and end (PEX, PEY) points which define the diagonal of the sub-pattern. Furthermore, a specific starting point for pattern RAM scanning is specified by PPX and PPY (see Figure 14).

Normally, the color registers (CL) should be loaded with on color data based on the number of bits per pixel. For example, if four bits/pixel are used, the 4-bit color pattern (e.g. 0001) should be replicated four times in the color register, as shown below.

Color Register =	0001	0001	0001	0001

In this way, color changes due to changing dot address are avoided.

AREA MODE

Prior to drawing, a drawing 'area' may be defined (area definition register). Then, during graphics drawing operation, the ACRTC will check if the drawing point is attempting to enter or exit the defined drawing area. Based on eight area modes, the ACRTC will take appropriate action for clipping or hitting.

---

* Normally, the color register (CL0 or CL1) selected by the pattern pointer (PPX, PPY) is used for the color data, but the source area data is used in the graphic copy commands (AGCPY and RGCPY).

** Normally, the color register (CL0 or CL1) selected by the pattern pointer (PPX, PPY) is used for the color register data (CL), but the source area data is used in the graphic copy command (AGCPY and RGCPY).

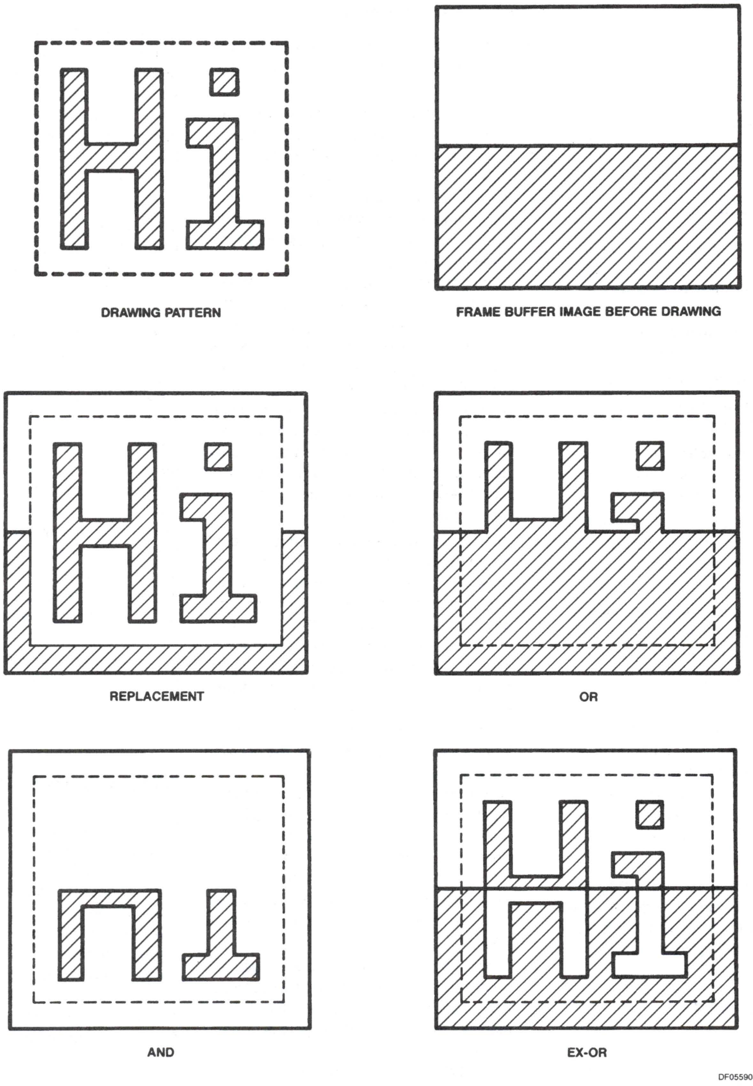

DRAWING PATTERN
FRAME BUFFER IMAGE BEFORE DRAWING
REPLACEMENT
OR
AND
EX-OR
DF05590S

Figure 13. Operation Mode Example

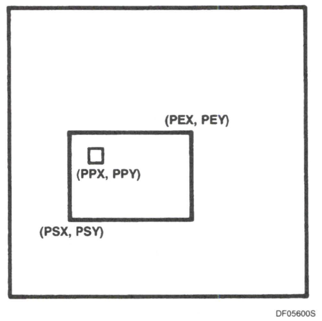


Figure 14. Pattern RAM

AREA	Drawing Area Mode
0 0 0	Drawing is executed without Area checking
0 0 1	When attempting to exit the area, drawing is stopped after setting ABT (abort bit)
0 1 0	Drawing suppressed outside the area — drawing operation continues and the ARD flag is not set
0 1 1	Drawing suppressed outside the area — drawing operation continues and the ARD flag is set at every drawing operation.
1 0 0	Same as area = 000
1 0 1	When attempting to enter the area, drawing is stopped after setting ABT (abort bit)
1 1 0	Drawing suppressed inside the area — drawing operation continues and the ARD flag is not set.
1 1 1	Drawing suppressed inside the area — drawing operation continues and the ARD flag is set at every drawing operation.

SYSTEM INTERFACE

Basic Clock

The ACRTC basic clock is 2CLK. 2CLK controls all primary ACRTC display and logic timing parameters. 2CLK, along with the specification of number of bits per logical pixel, the graphic address increment mode, and the display access mode, also determines the video data rate. The basic clock must be input, noting its cycle, max. and min. of "high" and "low" level width.

In any case, care must be taken not to stop the basic clock, fixing it at "high" or "low", nor to use 2CLK line in open state, which can destroy the LSI.

CRT INTERFACE

Frame Buffer Access

Access Modes — The three ACRTC display memory access modes are single, interleaved and superimposed.

1. Single access mode. A display (or drawing) cycle is defined as two cycles of 2CLK. During the first 2CLK cycle, the frame buffer display or drawing address is output. During the second 2CLK cycle, the frame buffer data is read (display cycles and/or drawing cycles) or written (drawing cycles).
   In this mode, display and drawing cycles contend for access to the frame buffer. The ACRTC allows the priority to be defined as display priority or drawing priority. If display priority, drawing cycles are only allowed to occur during horizontal/vertical flyback period. So, a 'flashless' display is obtained at the expense of slower drawing. If drawing priority, drawing may occur during display so high-speed drawing is obtained, however the display may flash.
2. Interleaved access mode (dual access mode 0). In this mode, display cycles and drawing cycles are interleaved. A display/drawing cycle is defined as four cycles of 2CLK. During the first 2CLK cycle, the frame buffer display address is output. During the second 2CLK cycle, the display data is read from the frame buffer. During the third 2CLK cycle, the frame buffer drawing address is output. During the fourth 2CLK cycle, the drawing data is read or written.
   Since there is no contention between display and drawing cycles, a 'flashless' display is obtained while maintaining full drawing speed. However, for a given configuration, frame buffer memory access time must be twice as fast as an equivalent single access mode configuration.
3. Superimposed access mode (dual access mode 1). In this mode, two separate logical screens are accessed during each display cycle. The display cycle is defined as four 2CLK cycles. During the first 2CLK cycle, the background (upper, base or lower) screen frame buffer address is output. During the second 2CLK cycle, the background screen display data is read. During the third 2CLK cycle, the window screen frame buffer address or the drawing frame buffer address is output. During the fourth 2CLK cycle, the window screen display or drawing data is read (display or drawing) or written (drawing). Note that the third and fourth cycles can be used for drawing (similar to interleaved mode) when these cycles are not used for window display.

Graphic Address Increment Mode (GAI) — During display operation, the ACRTC can be programmed to control the graphic display address in seven ways, including increment by 1, 2, 4, 8 and 16 words, 1 word every two display cycles and no increment.

NOTE:
The SCN63484 (R mask version) does not support 16-word increment mode.

Setting GAI to increment by 2, 4, 8 or 16 words per display cycle achieves linear increases in the video data rate; i.e. for a given configuration setting GAI to 2, 4, 8 or 16 words will achieve 2, 4, 8 or 16 times the video data rate corresponding to GAI = 1. This allows increasing the number of bits/logical pixel and logical pixel resolution while meeting the 2CLK maximum frequency constraint.

Table 7 shows the summary relationship between 2CLK, display access mode, graphic address increment, number of bits/logical pixel, memory access time and video data rate. The frame buffer cycle frequency (f_C) is shown by the following equation where:

f_V = Dot Clock
N = No. bits/logical pixel
D = Display access mode 1 for single access mode 2 for interleaved and superimposed access modes
A = Graphic address increment (½, 1, 2, 4, 8, 16)
$f_C = (f_V \times N \times D)/(A \times 16)$

DYNAMIC RAM REFRESH

When dynamic RAMs (DRAMs) are used for the frame buffer memory, the ACRTC can automatically provide DRAM refresh addressing. The ACRTC maintains an 8-bit DRAM refresh counter which is decremented on each frame buffer access. During $\overline{\text{HSYNC}}$ low, the ACRTC will output the sequential refresh addresses on MAD. The refresh address assignment depends on graphic address increment (GAI) mode as shown in Table 8. The ACRTC provides "0" output on the remaining address line of MAD and MA/RA.

DRAM refresh cycle timing must be factored into the determination of $\overline{\text{HSYNC}}$ low pulse width (HSW – specified in units of frame buffer memory cycles). If the horizontal scan rate if f_h (kHz), number of DRAM refresh cycles is N and the DRAM refresh cycle time is t_r (ms) then horizontal sync width (HSW) is specified by the following equation:

$$HWY \geqslant N \;/\; (t_r \times f_h)$$

Table 7. Graphic Address Increment Modes

DOT RATE ACCESS MODE		16MHz		32MHz		64MHz		128MHz	
Color No. (Bit/Pixel)	**Memory Cycle**	**S**	**D**	**S**	**D**	**S**	**D**	**S**	**D**
1	250ns	—	+½	+½	+1	+1	+2	+2	+4
	500ns	+½	+1	+1	+2	+2	+4	+4	+8
2	250ns	+½	+1	+1	+2	+2	+4	+4	+8
	500ns	+1	+2	+2	+4	+4	+8	+8	+16
4	250ns	+1	+2	+2	+4	+4	+8	+8	+16
	500ns	+2	+4	+4	+8	+8	+16	+16	—
8	250ns	+2	+4	+4	+8	+8	+16	+16	—
	500ns	+4	+8	+8	+16	+16	—	—	—
16	250ns	+4	+8	+8	+16	+16	—	—	—
	500ns	+8	+16	+16	—	—	—	—	—

For example, if the scan rate is 15.75kHz and the DRAMs have 128 refresh cycles of 2ms, HSW must be greater than or equal to 5.

$$HSW \geqslant 128 \ / \ (2 \times 15.75) = 4.06$$

EXTERNAL SYNCHRONIZATION

The ACRTC EXSYNC pin allows synchronization of multiple ACRTCs or other video signal generators. The ACRTC may be programmed as a single master device, or as one of a number of slave devices. To synchronize multiple ACRTCs, simply connect all the EXSYNC pins together.

For synchronizing to other video signals, the connection scheme depends on the raster scan mode. In non-interlace mode, EXSYNC corresponds to VSYNC. In interlace modes, EXSYNC corresponds to VSYNC of the odd field (see Figure 15).

NOTES:

1. The ACRTC performs the synchronization every time it accepts the pulse input from EXSYNC in the slave mode. It is recommended that the synchronous pulse should be input from EXSYNC only when the synchronization gap between the synchronous signal of the master device and that of ACRTC is in the slave mode. HSYNC and VSYNC are also output in the slave mode.
2. The ACRTC needs to be controlled not to execute the drawing operation during EXSYNC input.

Table 8. GAI and DRAM Refresh Addressing

ADDRESS INCREMENT MODE	REFRESH ADDRESS OUTPUT TERMINAL
+0 (GAI = 101)	MAD0 – MAD7
+1 (GAI = 000)	MAD0 – MAD7
+2 (GAI = 001)	MAD1 – MAD8
+4 (GAI = 010)	MAD2 – MAD9
+8 (GAI = 011)	MAD3 – MAD10
+16 (GAI = 100)	MAD4 – MAD11
+½ (GAI = 111)	MAD0 – MAD7
(GAI = 110)	

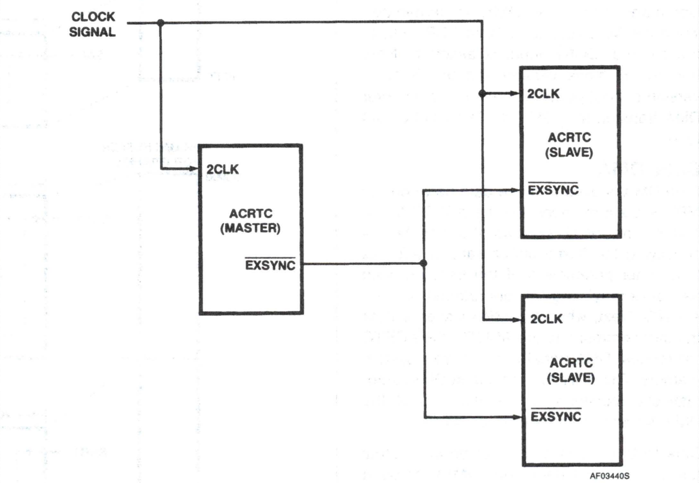


Figure 15. External Synchronization

MPU INTERFACE

MPU Bus Cycle

The ACRTC interfaces to the MPU as a peripheral occupying two addresses in the MPU address space. The ACRTC can operate as an 8- or 16-bit peripheral as configured during RES. An MPU bus cycle is initiated when CS is asserted (following the assertion of RS and R/W). The ACRTC responds to CS low by asserting DTACK low to complete the data transfer. DTACK will be returned to the MPU in between 1 and 1.5 2CLK cycles.

MPU WAIT states will be added in the following two cases.

1. If the ACRTC 2CLK input is much slower than the MPU clock, continuous ACRTC accesses may be delayed due to internal processing of the previous bus cycle. Be careful of CS "high" width.
2. If an ACRTC read cycle immediately follows an ACRTC write cycle, a wait state may occur due to ACRTC prepara-

tion for bus 'turn-around'. However, (e.g. 68000 system) MPUs normally have no instructions which immediately follow a write cycle with a read cycle.

For connection to synchronous bus interface MPUs, DTACK can simply be left open, assuming the system design guarantees that WAIT states cannot occur as described above. If WAIT states may occur, DTACK can be used with external logic to synthesize a READY signal.

DMA TRANSFER

The ACRTC can interface with an external DMA controller using three handshake signals. DMA request (DREQ), DMA acknowledge (DACK) and DMA done (DONE). The ACRTC uses the external DMAC for two types of transfers, command/parameter DMA and data DMA. For both types, DMA transfers use the ACRTC read and write FIFOs.

Command/Parameter DMA

The MPU initiates this mode by setting bit 12 (CDM) in the ACRTC command control register to 1. Then, the ACRTC will automatically request DMA transfer for commands and their associated parameters as long the write FIFO has space. Only cycle steal request mode (DREQ pulses low for each data transfer) can be used.

Command/parameter DMA is terminated when the MPU resets bit 12 in CCR to 0, or the external DONE input is asserted. Note that the R mask version and the S mask version cannot perform command/parameter DMA transfer, so CDM (bit 12) should be set to 0.

Data DMA

Data DMA is used to move data between the MPU system memory and the ACRTC frame buffer. The MPU sets up the transfer by specifying the frame buffer transfer address (and other parameters of the transfer, such as 'on-the-fly' logical operations) to the ACRTC. Next, when the MPU issues a data transfer command to the ACRTC, the ACRTC will request DMA transfer to and from system memory. The ACRTC will request DMA, automatically monitoring FIFO status, until the DMA transfer command is completed.

Data DMA request mode can be cycle steal (as in command/parameter DMA), or burst mode in which DREQ is a low-level control output to the DMAC which allows multiple data transfers during each acquisition of the MPU bus.

INTERRUPTS

The ACRTC recognizes eight separate conditions which can generate an interrupt, including command error detection, command end,

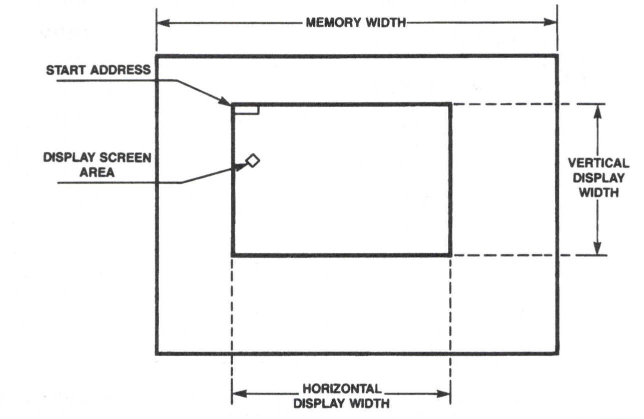

MEMORY WIDTH
START ADDRESS
DISPLAY SCREEN AREA
VERTICAL DISPLAY WIDTH
HORIZONTAL DISPLAY WIDTH
AF03450S

Figure 16. Display Screen/Frame Buffer Relationship

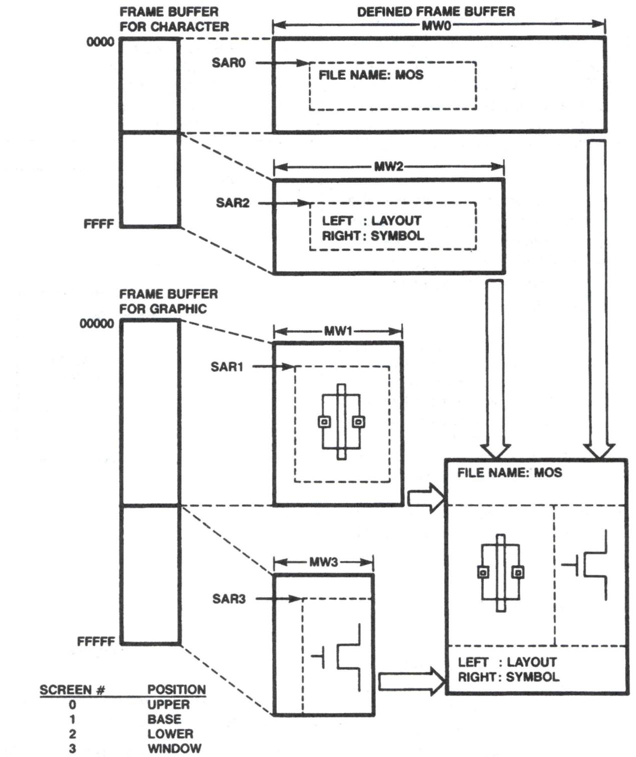

FRAME BUFFER FOR CHARACTER
DEFINED FRAME BUFFER
MW0
0000
SAR0
FILE NAME: MOS
MW2
SAR2
FFFF
LEFT : LAYOUT
RIGHT: SYMBOL
FRAME BUFFER FOR GRAPHIC
00000
MW1
SAR1
FILE NAME: MOS
MW3
SAR3
FFFFF
LEFT : LAYOUT
RIGHT: SYMBOL
LEFT : LAYOUT
RIGHT: SYMBOL
SCREEN #
POSITION
0 UPPER
1 BASE
2 LOWER
3 WINDOW
BD02530S

Figure 17. Display Screen Combination

drawing edge detection, light pen strobe, and four FIFO status conditions. Each condition has an associated mask bit for enabling/disabling the associated interrupt. The ACRTC removes the interrupt request when the MPU performs appropriate interrupt service by reading or writing to the ACRTC.

DISPLAY FUNCTION

Logical Display Screens

The ACRTC allows division of the frame buffer into four separate logical screens.

Screen Number	Screen Name	Screen Group Name
0	Upper Screen	Background Screens
1	Base Screen	
2	Lower Screen	
3	Window Screen	

In the simplest case, only the base screen parameters must be defined. Other screens may be selectively enabled, disabled and blanked under software control.

The background (upper, base, and lower) screens partition the display into three horizontal splits whose position is fully-programmable. A typical application might use the base screen for the bulk of user interaction, using the lower screen for 'status line(s)' and the upper screen for 'pull-down menu(s)'.

The window screen is unique, since the ACRTC gives the window screen higher priority than background screens. Thus, when the window, whose size and position is fully-programmable, overlaps a background screen, the window screen is displayed. One exception is the ACRTC superimposed access mode, in which the window has the same display priority as background screens. In this case, the window and background screen are 'superimposed' on the display. The ACRTC logical screen organization can be programmed to best suit a number of display applications. See Figures 16 through 18.

GRAPHIC/CHARACTER ADDRESS SPACES

The ACRTC controls two separate logical address spaces. The CHR pin allows external decoding if physically-separate frame buffers are desired. Each of the four logical screens (upper, base, lower, and window) is programmed as residing in the graphics address space or the character address space (see Figure 19).

ACRTC accesses to graphics screens are treated as bit-mapped using a 20-bit frame buffer address, with an address space of one megaword (1M by 16 bits).

ACRTC accesses to character screens are treated as character generator-mapped. In this case, a 64k word address space is used and five bits of raster address are output to an external character generator (see Figure 20).

Multiple logical screens defined as character can be externally decoded to use separate character generators or different addresses within a combined character generator. Also, each character screen may be defined with separate line spacing, separate cursors, etc.

CURSOR CONTROL

The ACRTC has two block cursor registers and a graphics cursor register. A block cursor is used with character screens. The cursor start and ending raster addresses are fully-programmable. Also, the cursor width can be defined as one to eight memory cycles.

A graphics cursor is defined by specifying the start/end memory in cycle the X dimension and the start/end raster in the Y dimension. The graphic cursor can output on character screens. The ACRTC provides two separate cursor outputs, $\overline{\text{CUD1}}$ and $\overline{\text{CUD2}}$. These are combined with two character cursor registers and a graphics cursor register to provide three cursor modes.

Block Mode

Two block cursors are output on $\overline{\text{CUD1}}$ and $\overline{\text{CUD2}}$, respectively (see Figures 21 and 22).

Graphic Mode

The graphic cursor is output on CUD1. Using an external cursor pattern memory allows a graphic cursor of various shapes. Two block cursors are multiplexed on CUD2 (see Figure 23).

Crosshair Mode

The horizontal and vertical components of the graphic cursor are output on CUD1 and CUD2, respectively. This allows simple generation of a crosshair cursor control signal (see Figure 24).

SCROLLING

Vertical Scroll

Each logical screen performs independent vertical scroll. On character screens, vertical smooth scroll is accomplished using the programmable start address raster (SAR). Line-by-line scroll is accomplished by increasing or decreasing the screen start address by one unit of horizontal memory width.

On graphics screens, vertical smooth scroll is accomplished by increasing or decreasing the screen start address by one unit of horizontal memory width (see Figure 25).

Horizontal Scroll

Horizontal scroll can be performed in units of characters for character screens and units of words (multi-logical pixels) for graphic screens by increasing or decreasing the screen start address by 1. For smooth horizontal scroll, the ACRTC has dot shift video attributes which can be used with an external circuit which conditions shift register load/clocking.

Since this dot shift information is output each raster, horizontal smooth scroll is limited to either the background screens or the window screen at any given time. However, horizontal smooth scroll is independent for each of the background screens (upper, base, lower). See Figures 25 through 27.

RASTER SCAN MODES

The ACRTC has three software-selectable raster scan modes — non-interlace, interlace sync, and interlace sync and video. In non-interlace mode, a frame consists of one field. In the interlace modes, a frame consists of two fields, the even and odd fields.

The interlace modes allow increasing screen resolution while avoiding limits imposed by the CRT display device, such as maximum horizontal scan frequency or maximum video dot rate.

Interlace sync mode simply repeats each raster address for both the even and odd fields. This is useful for increasing the quality of a displayed figure when using an interlaced CRT device such as a television set with RF modulator.

Interlace sync and video mode displays alternate even and odd rasters on alternate even and odd fields. For a given number of rasters/character, this mode allows twice as many characters to be displayed in the vertical direction as non-interlace mode (see Figure 28).

Note that for interlace modes, the refresh frequency for a given dot on the screen is one-half that of the non-interlace mode. Interlace modes normally require the use of a CRT with a more persistent phosphor to avoid flickering display.

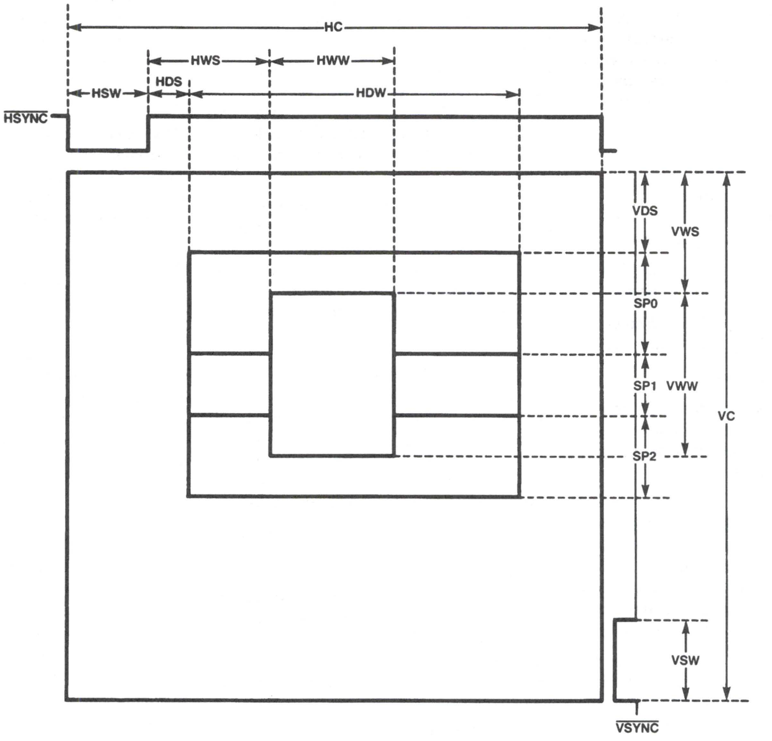

HC
HWS
HWW
HDS
HSW
HDW
HSYNC
VDS
VWS
SP0
SP1
VWW
VC
SP2
VSW
VSYNC
AF03460S

Figure 18. Display Screen Specification

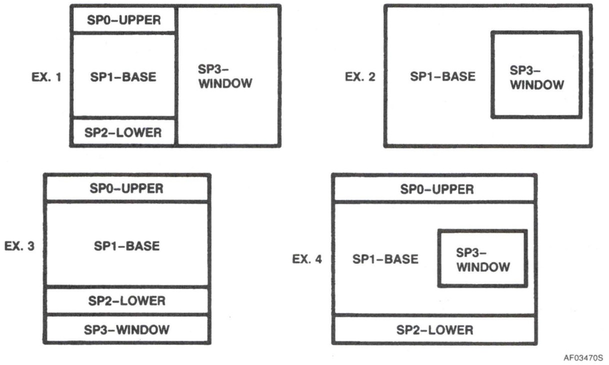

EX. 1
SP0-UPPER
SP1-BASE
SP2-LOWER
SP3-WINDOW
EX. 2
SP1-BASE
SP3-WINDOW
EX. 3
SP0-UPPER
SP1-BASE
SP2-LOWER
SP3-WINDOW
EX. 4
SP0-UPPER
SP1-BASE
SP3-WINDOW
SP2-LOWER
AF03470S

Figure 19. Screen Combination Example

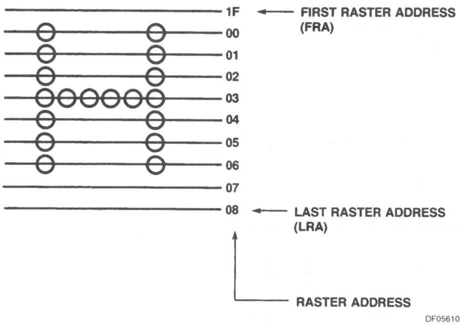

1F
00
01
02
03
04
05
06
07
08
FIRST RASTER ADDRESS (FRA)
LAST RASTER ADDRESS (LRA)
RASTER ADDRESS
DF05610S

Figure 20. Character Screen Raster Addressing

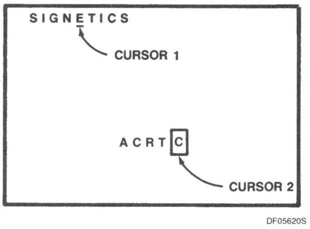

SIGNETICS
CURSOR 1
ACRTC
CURSOR 2
DF05620S

Figure 21. Two Separate Block Cursors

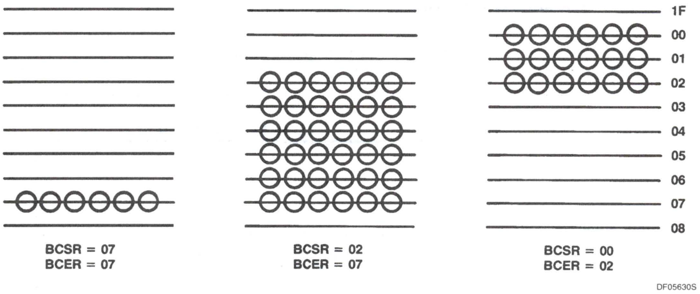

BCSR = 07
BCER = 07
BCSR = 02
BCER = 07
BCSR = 00
BCER = 02
1F
00
01
02
03
04
05
06
07
08
DF05630S

Figure 22. Block Cursor Examples

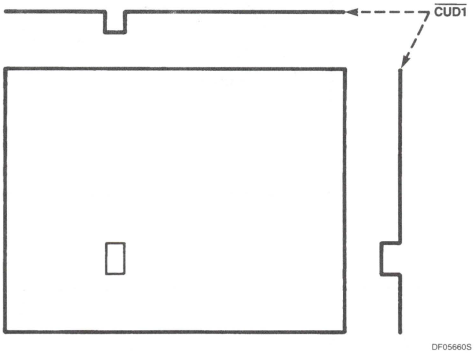

$\overline{\text{CUD1}}$
DF05660S

Figure 23. Graphic Cursor

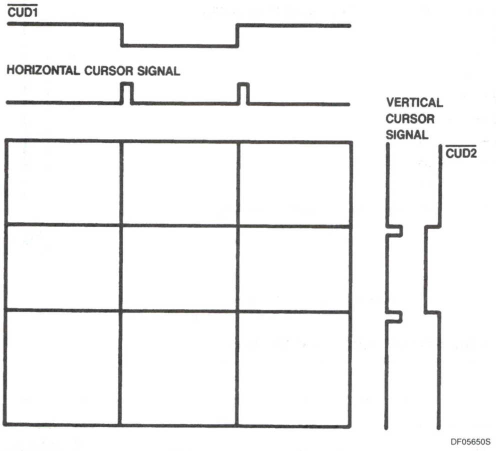

$\overline{\text{CUD1}}$
HORIZONTAL CURSOR SIGNAL
VERTICAL CURSOR SIGNAL
$\overline{\text{CUD2}}$
DF05650S

Figure 24. Crosshair Cursor

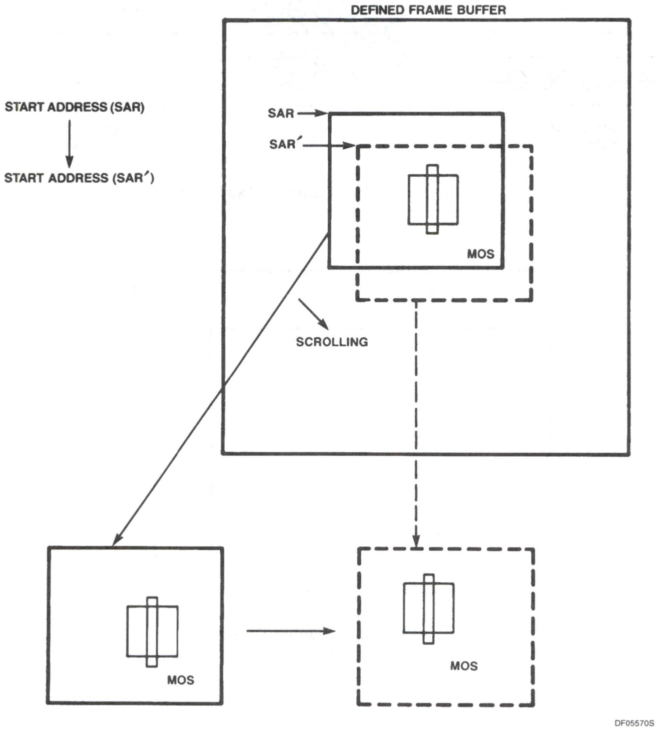

DEFINED FRAME BUFFER
START ADDRESS (SAR)
START ADDRESS (SAR′)
SAR
SAR′
MOS
SCROLLING
MOS
MOS
DF05570S

Figure 25. Scrolling By SAR (Start Address Register) Rewrite

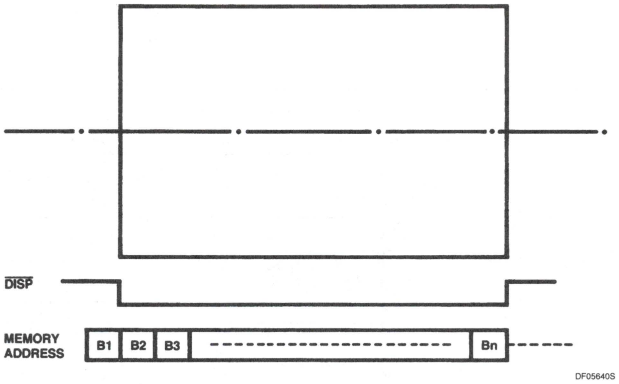

DISP
MEMORY ADDRESS
B1
B2
B3
Bn
DF05640S

Figure 26. Horizontal Smooth Scroll — Base Screen

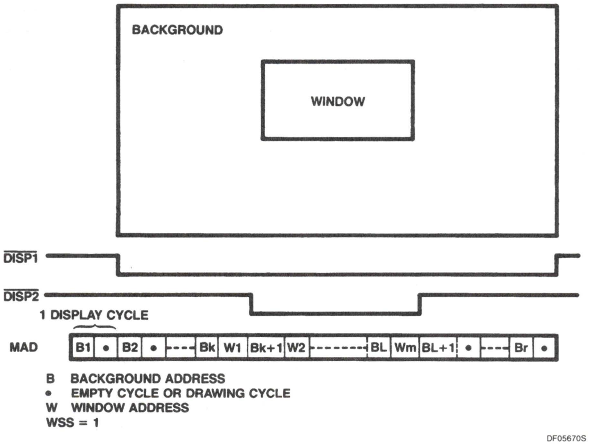

BACKGROUND
WINDOW
DISP1
DISP2
1 DISPLAY CYCLE
MAD
B1
•
B2
•
Bk
W1
Bk+1
W2
BL
Wm
BL+1
•
Br
•
B BACKGROUND ADDRESS
• EMPTY CYCLE OR DRAWING CYCLE
W WINDOW ADDRESS
WSS = 1
DF05670S

Figure 27. Horizontal Smooth Scroll — Window Screen

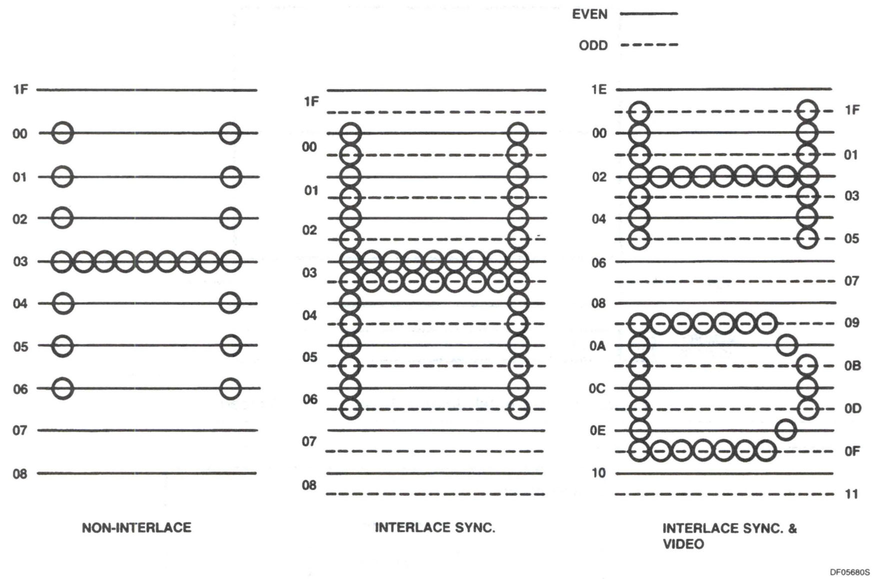

EVEN
ODD
1F
00
01
02
03
04
05
06
07
08
NON-INTERLACE
INTERLACE SYNC.
1E
1F
00
01
02
03
04
05
06
07
08
09
0A
0B
0C
0D
0E
0F
10
11
INTERLACE SYNC. & VIDEO
DF05680S

Figure 28. Raster Screen Modes

ZOOMING

The base screen (screen 1) is supported by the ACRTC zooming function. Note that ACRTC zooming is performed by controlling the CRT timing signals. The contents of the frame buffer area being zoomed are not changed. The ACRTC allows specification of a zoom factor (1 to 16) independently in the X and Y directions.

For horizontal zoom, the programmed zoom factor is output as video attributes. An external circuit uses this factor to condition the external shift register clock to accomplish horizontal zooming.

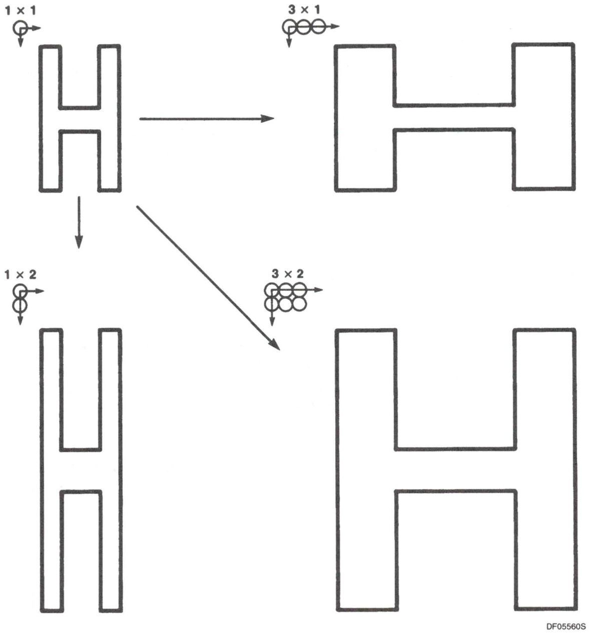
For vertical zoom, no external circuit is required. The ACRTC will scan a single raster multiple times to accomplish vertical zooming (see Figure 29).

LIGHT PEN

The ACRTC provides a 20-bit light pen address register and a light pen strobe (DPSTB) input pin for connection with a light pen.

A light pen strobe pulse will occur when the CRT electron beam passes under the light pen during display refresh. When this pulse occurs, the contents of the ACRTC display refresh address counter will be latched into the light pen address register along with a logical screen (character or graphic screen) designator. Also, an ACRTC status flag indicating light pen activity is set, generating an optional (maskable) MPU interrupt. Note that for superimposed access mode, when the light pen strobe occurs in an area in which the window overlaps a background (upper, base, or lower) screen, the background screen address will be latched. And even for all access modes, the drawing address will be latched.

Various system and ACRTC delays will cause the latched address to differ slightly from the actual light pen position. The light pen address can be corrected using software, based on system-specific delays. Or, if the application does not require the highest light pen pointing resolution, software can 'bind' the light pen address by specifying a range of values associated with a given area of the screen.



NOTES:
1. "U" mask limitation:
2. — when the parameters "a, b, and DX" for the ellipse command satisfy the following condition, a ⩾ 2b(2DX – 1), the command fails to end and the ACRTC goes into a deadlock, resulting in a Flat Ellipse.

Figure 29. Zooming

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
V_S	Supply voltage[2] range	−0.3 to +7.0	V
V_{IN}	Input voltage[2] range	−0.3 to V_{CC} + 0.3	V
I_{OUT}	Allowable output current[3]	+5	mA
I_{OUT}	Total allowable output current[4]	+120	mA
T_A	Operating temperature range	0 to +70	°C
T_{STG}	Storage temperature range	−55 to +150	°C

RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
V_S	Supply voltage[2]	4.75	5	5.25	V
V_{IL}	Input low level voltage[2]	0	–	0.7	V
V_{IH}	Input high level voltage[2]	2.2	–	V_{CC}	V
T_A	Operating temperature	0	25	70	°C

NOTES:

1. Using an LSI beyond its maximum ratings may result in permanent destruction. LSIs should usually be used under recommended operating conditions. Exceeding any of these conditions may adversely affect the LSI's reliability.
2. This value is in reference to V_{SS} = 0V.
3. The allowable output current is the maximum current that may be drawn from, or flow out to one output terminal or one input/output common terminal.
4. The total allowable output current is the total sum of currents that may be drawn from, or flow out to output terminals or input/output common terminals.

TIMING MEASUREMENT

The timing measurement point for the output "low" level is defined at 0.8V throughout this specification. The output "low" level at stable condition (DC characteristics) is defined at 0.5V. The output "high" level is defined at V_{CC} − 2.0V (see Figure 30).

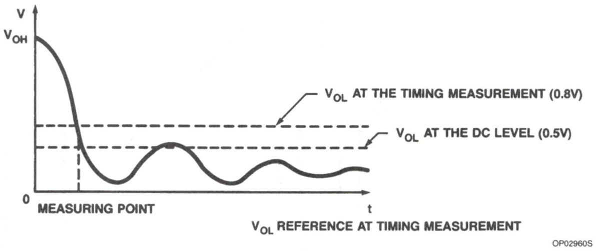

V
V_{OH}
V_{OL} AT THE TIMING MEASUREMENT (0.8V)
V_{OL} AT THE DC LEVEL (0.5V)
0
MEASURING POINT
t
V_{OL} REFERENCE AT TIMING MEASUREMENT
OP02960S

Figure 30. Timing Measurement

DC ELECTRICAL CHARACTERISTICS V_{CC} = 5V ±5%, V_{SS} = 0V, T_A = 0 to +70°C unless otherwise noted.

PARAMETER		TEST CONDITIONS	8MHz		UNIT
			Min	Max	
Input high level	All inputs		2.2	V_{CC}	V
Input low level	All inputs		−0.3	0.7	V
Input leakage current	R/W, $\overline{\text{CS}}$, RS, $\overline{\text{RES}}$, $\overline{\text{DACK}}$ 2CLK, LPSTB	V_{IN} = 0 − V_{CC}	−2.5	2.5	μA
3-State (off-state) input current	D0 − D15, $\overline{\text{EXSYNC}}$, MAD0 − MAD15	V_{IN} = 0.4 − V_{CC}	−10	10	μA
Output high level	D0 − D15, MAD0 − MAD15, $\overline{\text{CUD1}}$, $\overline{\text{CUD2}}$ $\overline{\text{DREQ}}$, $\overline{\text{DTACK}}$, $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, $\overline{\text{EXSYNC}}$	I_{OH} = −400μA	V_{CC} −1	—	V
Output low level	$\overline{\text{DISP1}}$, $\overline{\text{DISP2}}$ CHR, MRD, $\overline{\text{DRAW}}$, $\overline{\text{AS}}$, MCYC, RA4, MA16/RA0, MA19/RA3	I_{OL} = 2.2mA	—	0.5	V
Output leakage current (off-state)	$\overline{\text{IRQ}}$, $\overline{\text{DONE}}$	V_{OH} = V_{CC}	—	10	μA
Input capacity	D0 − D15, $\overline{\text{EXSYNC}}$, MAD0 − MAD15 R/$\overline{\text{W}}$, $\overline{\text{CS}}$, RS, $\overline{\text{RES}}$, $\overline{\text{DACK}}$, 2CLK, LPSTB	V_{IN} = 0V, T_A = 25°C, f = 1.0MHz	—	17	pF
Output capacity	$\overline{\text{IRQ}}$, $\overline{\text{DONE}}$	V_{IN} = 0V, T_A = 25°C, f = 1.0MHz	—	15	pF
Current Consumption		Chip not selected Display in progress	—	100	mA
		Data bus in read/write operation Display in progress Command execution in progress	—	100	mA

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, $V_{SS} = 0V$, $T_A = 0$ to $+70°C$ unless otherwise noted.

NO.	FIGURE	CHARACTERISTIC	TENTATIVE LIMITS 8MHz Min	TENTATIVE LIMITS 8MHz Max	UNIT
		Operation frequency of 2CLK	1	8	MHz
1	41	Clock cycle time	125	1000	ns
2	41	Clock high level pulse width	55	500	ns
3	41	Clock low level pulse width	55	500	ns
4	41	Clock rise time	—	10	ns
5	41	Clock fall time	—	10	ns
6	42, 43	R/$\overline{\text{W}}$ setup time	50	—	ns
7	42, 43	R/$\overline{\text{W}}$ hold time	0	—	ns
8	42, 43	RS setup time	50	—	ns
9	42, 43	RS hold time	0	—	ns
10	42, 43	$\overline{\text{CS}}$ setup time	40	—	ns
11	42, 43	$\overline{\text{CS}}$ high level width	60	—	ns
13	42	Read wait time	0	—	ns
14	42	Read data access time	—	80	ns
15	42	Read data hold time	10	—	ns
16	42	Read data turn off time	—	60	ns
17	42, 43	$\overline{\text{DTACK}}$ delay time (Z to L)	—	70	ns
18	42	$\overline{\text{DTACK}}$ delay time (D to L)	0	—	ns
19	42, 43	$\overline{\text{DTACK}}$ release time (L to H)	—	80	ns
20	42, 43	$\overline{\text{DTACK}}$ turn off time (H to Z)	—	100	ns
21	42	Data bus 3-State recovery time 1	0	—	ns
22	43	Write wait time	0	—	ns
23	43	Write data setup time	40	—	ns
24	43	Write data hold time	10	—	ns
25	44, 45	$\overline{\text{DREQ}}$ delay time 1	—	110	ns
26	44, 45	$\overline{\text{DREQ}}$ delay time 2	—	70	ns
27	44, 45	DMA R/$\overline{\text{W}}$ setup time	50	—	ns
28	44, 45	DMA R/$\overline{\text{W}}$ hold time	0	—	ns
29	44, 45	$\overline{\text{DACK}}$ setup time	40	—	ns
30	44, 45	$\overline{\text{DACK}}$ high level width	60	—	ns
32	44	DMA read wait time	0	—	ns
33	44	DMA read data access time	—	80	ns
34	44	DMA read data hold time	10	—	ns
35	44	DMA read data turn off time	—	60	ns
36	44, 45	DMA $\overline{\text{DTACK}}$ delay time (Z to L)	—	70	ns
37	44	DMA $\overline{\text{DTACK}}$ delay time (D to L)	0	—	ns
38	44, 45	DMA $\overline{\text{DTACK}}$ release time (L to H)	—	80	ns
39	44, 45	DMA $\overline{\text{DTACK}}$ turn off time (H to Z)	—	100	ns
40	44, 45	$\overline{\text{DONE}}$ output delay time	—	70	ns

AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	FIGURE	CHARACTERISTIC	TENTATIVE LIMITS		UNIT
			8MHz		
			Min	Max	
41	44, 45	$\overline{\text{DONE}}$ output turn off time (L to Z)	—	80	ns
42	44	Data bus 3-State recovery time 2	0	—	ns
43	44, 45	$\overline{\text{DONE}}$ input pulse width	2	—	Clk Cyc
44	45	DMA write wait time	0	—	ns
45	45	DMA write data setup time	40	—	ns
46	45	DMA write data hold time	10	—	ns
48	46 – 49	$\overline{\text{AS}}$ low level pulse width	25	—	ns
49	47, 48	Memory address hold time 2	10	—	ns
50	46 – 49	$\overline{\text{AS}}$ delay time 1	—	60	ns
51	46 – 49	$\overline{\text{AS}}$ delay time 2	—	60	ns
52	46 – 49	Memory address delay time	—	70	ns
53	46 – 49	Memory address hold time 1	10	—	ns
54	46, 47, 49	Memory address turn off time (A to Z)	—	50	ns
55	47	Memory address data setup time	40	—	ns
56	47	Memory read data hold time	10	—	ns
57	46 – 49	MA/RA delay time	—	80	ns
58	46 – 48	MA/RA hold time	10	—	ns
59	46 – 50	MCYC delay time	—	50	ns
60	46 – 49	MRD delay time	—	70	ns
61	46 – 49	MRD hold time	10	—	ns
62	46 – 49	$\overline{\text{DRAW}}$ delay time	—	70	ns
63	46 – 49	$\overline{\text{DRAW}}$ hold time	10	—	ns
64	48	Memory write data delay time	—	70	ns
65	48	Memory write data hold time	10	—	ns
67	49 – 51	$\overline{\text{HSYNC}}$ delay time	—	70	ns
68	50	$\overline{\text{VSYNC}}$ delay time	—	70	ns
69	50	$\overline{\text{DISP1}}$, $\overline{\text{DISP2}}$ delay time	—	70	ns
70	50	$\overline{\text{CUD1}}$, $\overline{\text{CUD2}}$ delay time	—	70	ns
71	50	$\overline{\text{EXSYNC}}$ output delay time	20	70	ns
72	50	CHR delay time	—	70	ns
75	51	$\overline{\text{EXSYNC}}$ input pulse width	3	—	Clk Cyc
76	51	$\overline{\text{EXSYNC}}$ input setup time 1	50	—	ns
77	51	$\overline{\text{EXSYNC}}$ input hold time	30	—	ns
78	52	LPSTB uncertain time 1	70	—	ns
79	52	LPSTB uncertain time 2	10	—	ns
80	52	LPSTB input hold time	10	—	ns
81	52	LPSTB input inhibit time	4	—	Clk Cyc
82	53	$\overline{\text{DACK}}$ setup time for $\overline{\text{RES}}$	100	—	ns
83	53	$\overline{\text{DACK}}$ hold time for $\overline{\text{RES}}$	0	—	ns

Advanced CRT Controller (ACRTC) SCC63484

AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	FIGURE	CHARACTERISTIC	TENTATIVE LIMITS 8MHz Min	TENTATIVE LIMITS 8MHz Max	UNIT
84	53	$\overline{\text{RES}}$ input pulse width	10	—	Clk Cyc
85	54	$\overline{\text{IRQ}}$ delay time 1	—	150	ns
86	54	$\overline{\text{IRQ}}$ delay time 2	—	500	ns
87	49	ATR delay time 1	—	80	ns
88	49	ATR hold time 1	10	—	ns
90	49	ATR delay time 2	—	80	ns
91	49	ATR hold time 2	10	—	ns
100	42, 43	$\overline{\text{CS}}$ cycle time	4	—	Clk Cyc
101	42, 43	$\overline{\text{CS}}$ low level width	2	—	Clk Cyc
102	42, 43	$\overline{\text{CS}}$ high level width	2	—	Clk Cyc
104	44 45	$\overline{\text{DACK}}$ cycle time	4	—	Clk Cyc
105	44, 45	$\overline{\text{DACK}}$ low level width	2	—	Clk Cyc
106	44, 45	$\overline{\text{DACK}}$ high level width	2	—	Clk Cyc

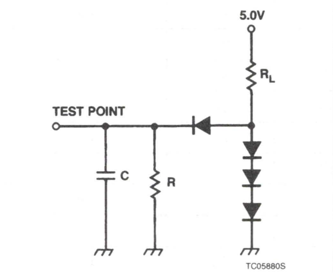

5.0V
R_L
TEST POINT
C
R
TC05880S

Figure 31. Test Load Circuit A

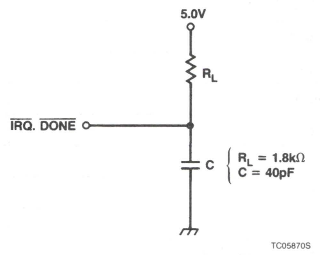

5.0V
R_L
$\overline{\text{IRQ}}$. $\overline{\text{DONE}}$
C
R_L = 1.8kΩ
C = 40pF
TC05870S

Figure 32. Test Load Circuit B

Signal	Load Condition
D0 – D15	R_L = 1.84Ω
$\overline{\text{DTACK}}$	C = 40pF
$\overline{\text{DREQ}}$	
MAD0 – MAD15	
MA16/RA0 – MA19/RA3	
RA4	R = 10kΩ
$\overline{\text{VSYNC}}$, $\overline{\text{HSYNC}}$	
$\overline{\text{EXSYNC}}$	All diodes are 1S2074 or equivalent
MCYC, $\overline{\text{AS}}$, MRD	
$\overline{\text{DRAW}}$, CHR	
$\overline{\text{DISP1}}$, $\overline{\text{DISP2}}$	
$\overline{\text{CUD1}}$, $\overline{\text{CUD2}}$	

Power-On Sequence

The following condition needs to be satisfied when the power turns on.

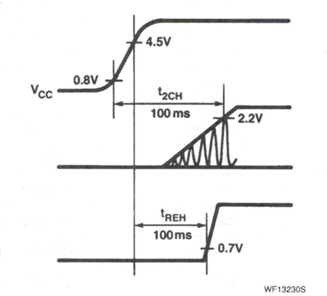

4.5V
V_{CC}
0.8V
t_{2CH}
100 ms
2.2V
t_{REH}
100ms
0.7V
WF13230S

Figure 33. Power-On Sequence

Output Waveform

In case that ringing noise occurs beyond tolerance on the CRT data buses (MAD0 – MAD15, MA16/RA0 – MA19/RA2, RA4), damping resistors may be required for data buses as shown in Figures 34 and 35.

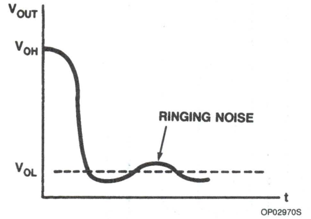


Figure 34. Ringing Noise

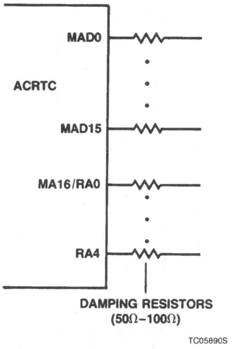

MAD0
ACRTC
MAD15
MA16/RA0
RA4
DAMPING RESISTORS
(50Ω–100Ω)
TC05890S

Figure 35. Damping Resistors

Power Supply Circuit

When designing the V_{CC} and V_{SS} pattern of the circuit board, the capacitors need to be located nearest to pin 14 (V_{CC}) and pin 16 (V_{SS}) or pin 51 (V_{SS}) and pin 49 (V_{CC}), as shown in Figure 36.

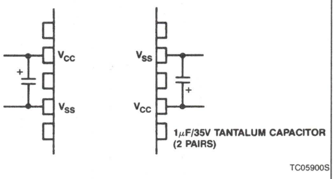

V_{CC}
V_{SS}
+
V_{SS}
V_{CC}
+
1μF/35V TANTALUM CAPACITOR
(2 PAIRS)
TC05900S

Figure 36. Designing Power Supply Current

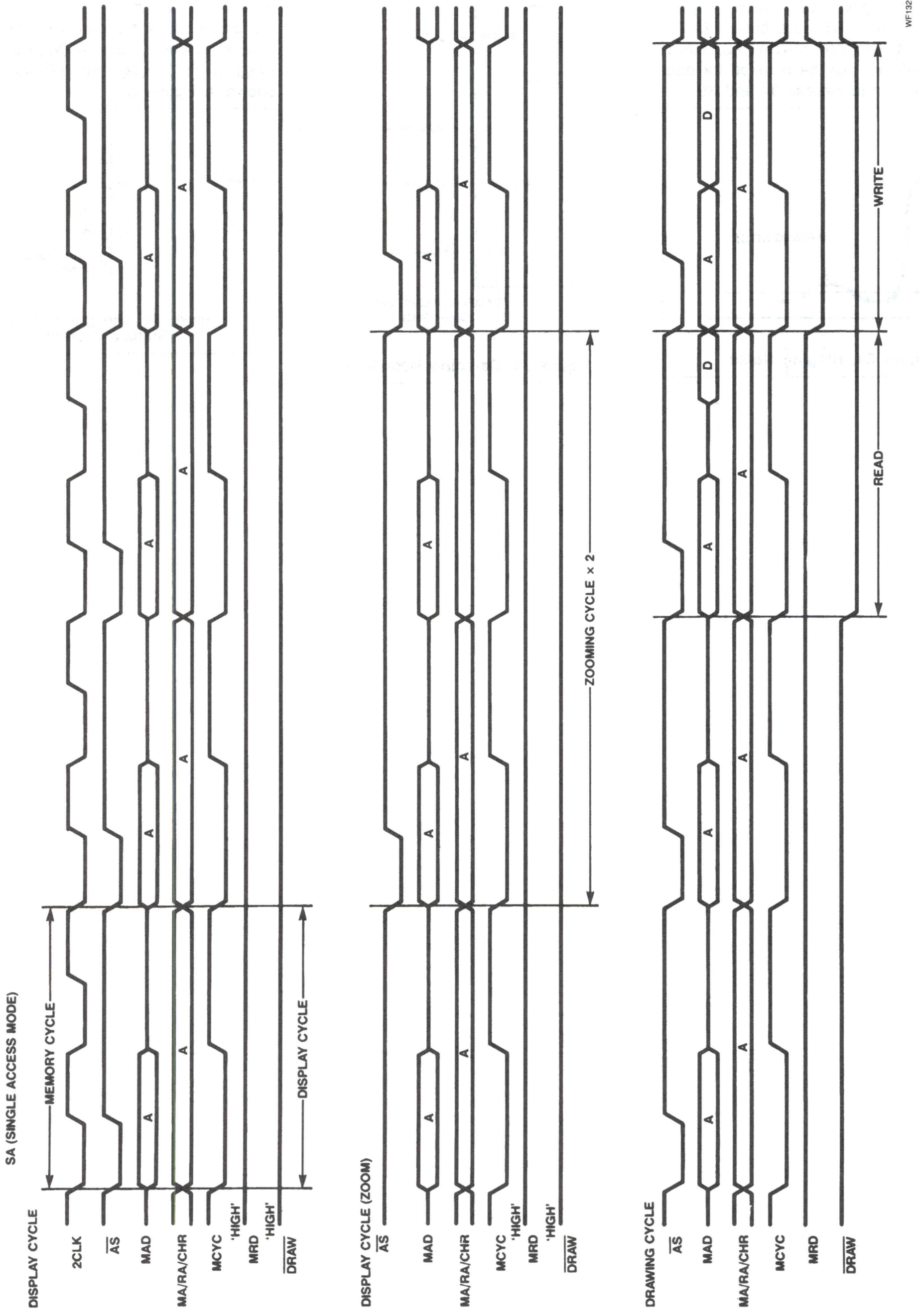

SA (SINGLE ACCESS MODE)
DISPLAY CYCLE
2CLK
AS
MAD
MA/RA/CHR
MCYC
'HIGH'
MRD
'HIGH'
DRAW
MEMORY CYCLE
DISPLAY CYCLE
A
DISPLAY CYCLE (ZOOM)
ZOOMING CYCLE × 2
DRAWING CYCLE
D
READ
WRITE
WF13250S

Figure 37. Access Mode Timing

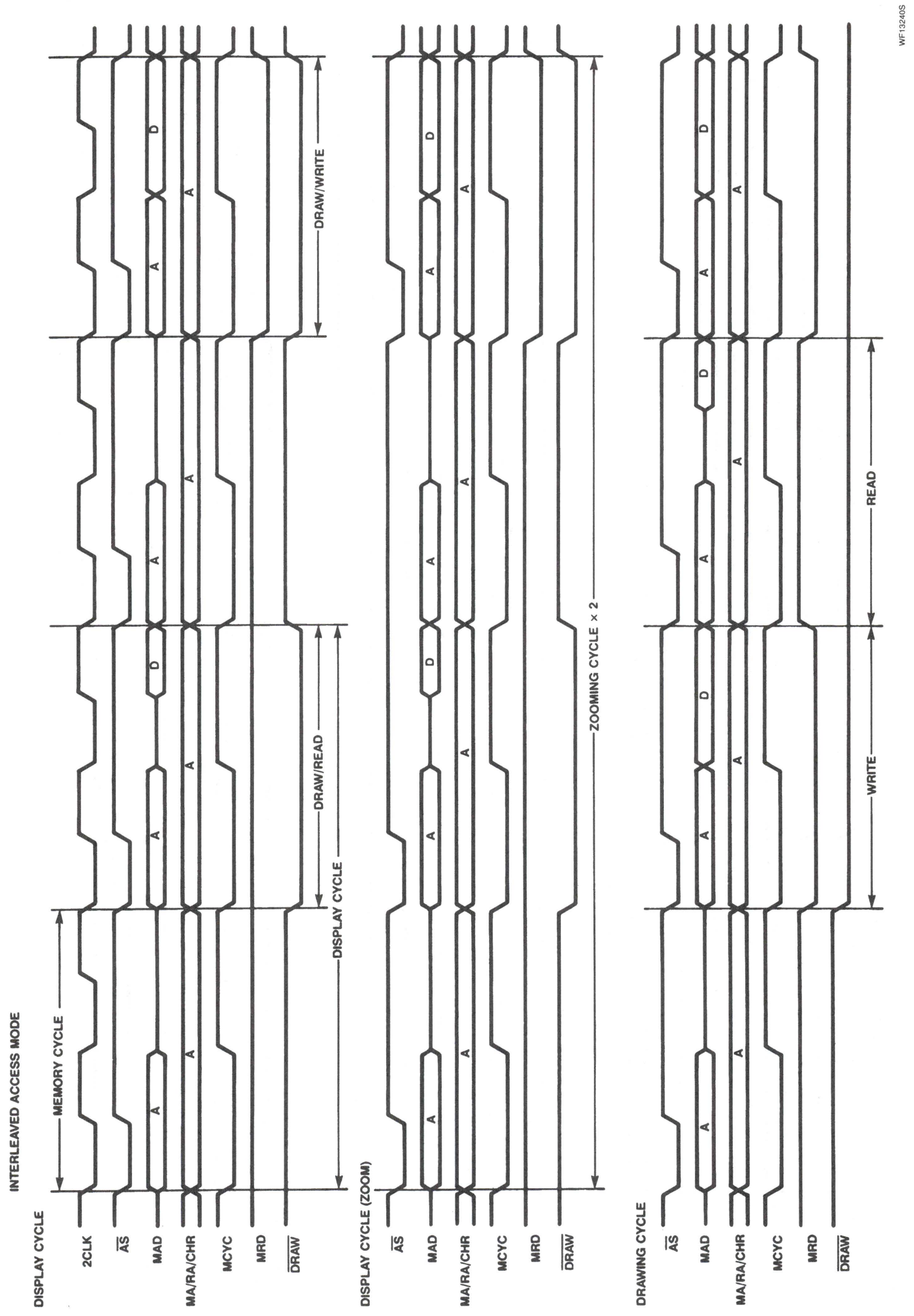

WF13240S
INTERLEAVED ACCESS MODE
DISPLAY CYCLE
2CLK
AS
MAD
MA/RA/CHR
MCYC
MRD
DRAW
MEMORY CYCLE
DISPLAY CYCLE
DRAW/READ
DRAW/WRITE
A
D
DISPLAY CYCLE (ZOOM)
AS
MAD
MA/RA/CHR
MCYC
MRD
DRAW
ZOOMING CYCLE × 2
DRAWING CYCLE
AS
MAD
MA/RA/CHR
MCYC
MRD
DRAW
WRITE
READ

Figure 37. Access Mode Timing (Continued)

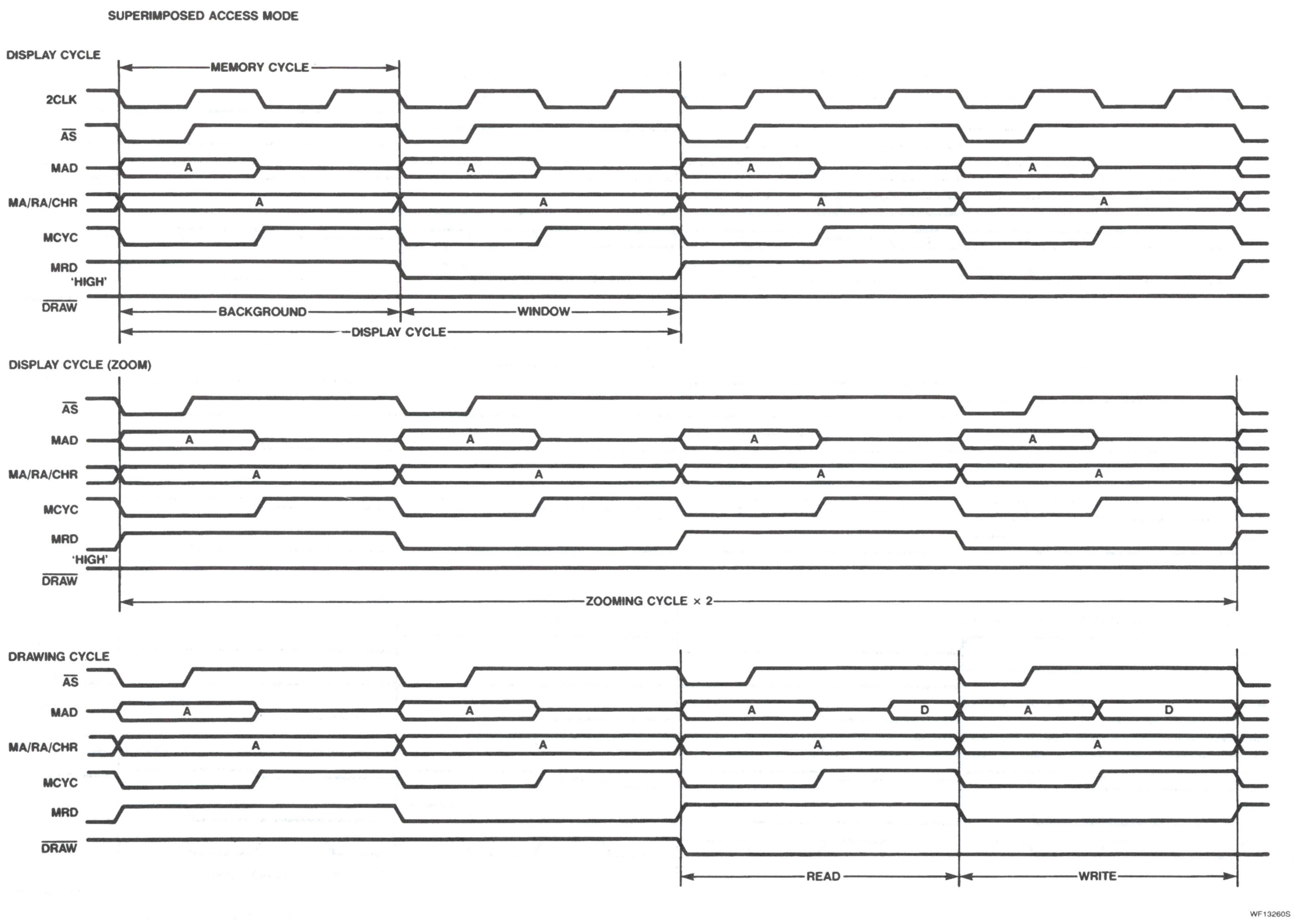

SUPERIMPOSED ACCESS MODE
DISPLAY CYCLE
MEMORY CYCLE
2CLK
AS
MAD
A
MA/RA/CHR
MCYC
MRD
'HIGH'
DRAW
BACKGROUND
WINDOW
DISPLAY CYCLE
DISPLAY CYCLE (ZOOM)
ZOOMING CYCLE × 2
DRAWING CYCLE
D
READ
WRITE
WF13260S

Figure 37. Access Mode Timing (Continued)

Advanced CRT Controller (ACRTC) SCC63484

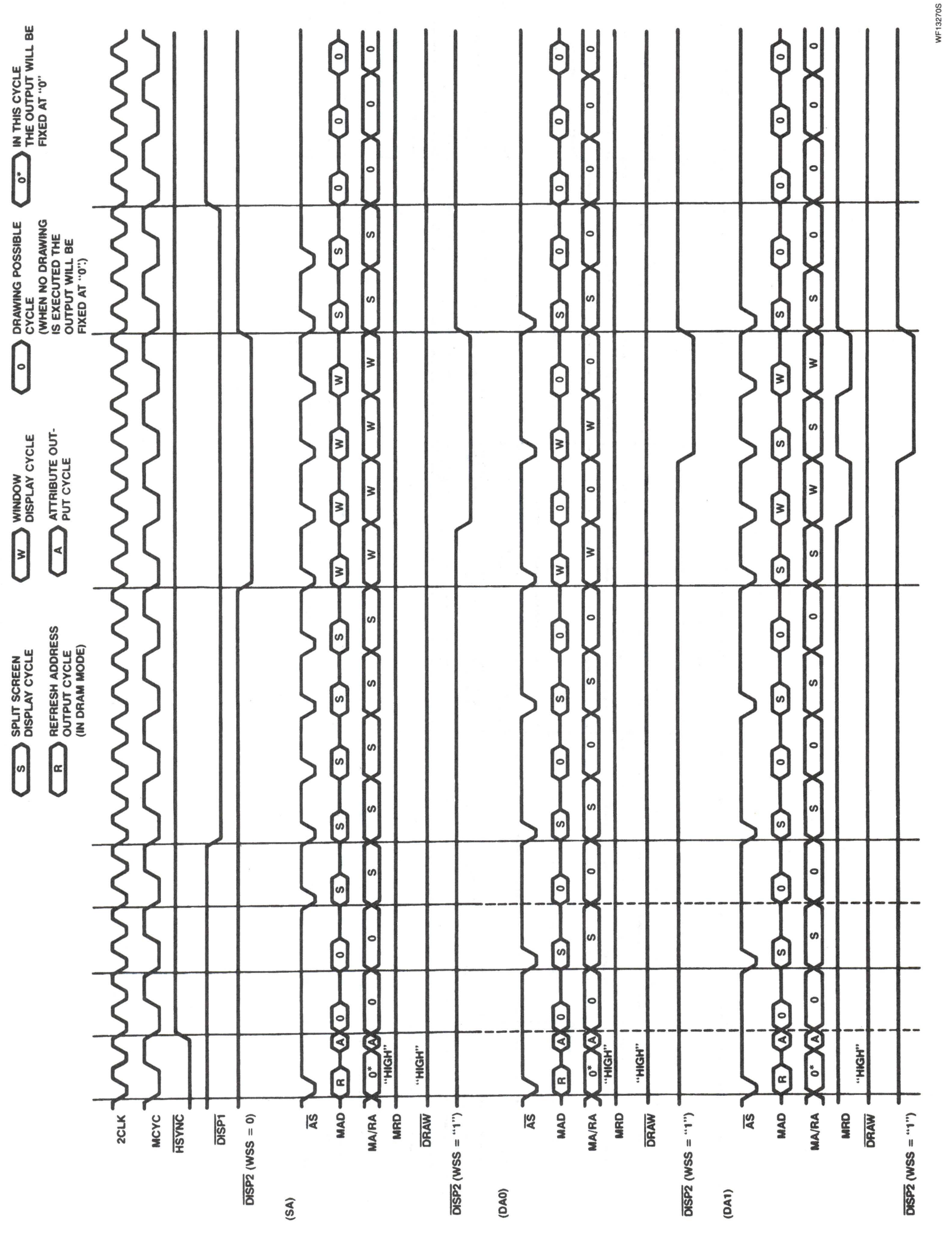

S SPLIT SCREEN DISPLAY CYCLE
R REFRESH ADDRESS OUTPUT CYCLE (IN DRAM MODE)
W WINDOW DISPLAY CYCLE
A ATTRIBUTE OUTPUT CYCLE
0 DRAWING POSSIBLE CYCLE (WHEN NO DRAWING IS EXECUTED THE OUTPUT WILL BE FIXED AT "0")
0* IN THIS CYCLE THE OUTPUT WILL BE FIXED AT "0"
2CLK
MCYC
HSYNC
DISP1
DISP2 (WSS = 0)
(SA)
AS
MAD
MA/RA
MRD
DRAW
DISP2 (WSS = "1")
(DA0)
AS
MAD
MA/RA
MRD
DRAW
DISP2 (WSS = "1")
(DA1)
AS
MAD
MA/RA
MRD
DRAW
DISP2 (WSS = "1")
"HIGH"
WF13270S

Figure 37. Access Mode Timing (Continued)

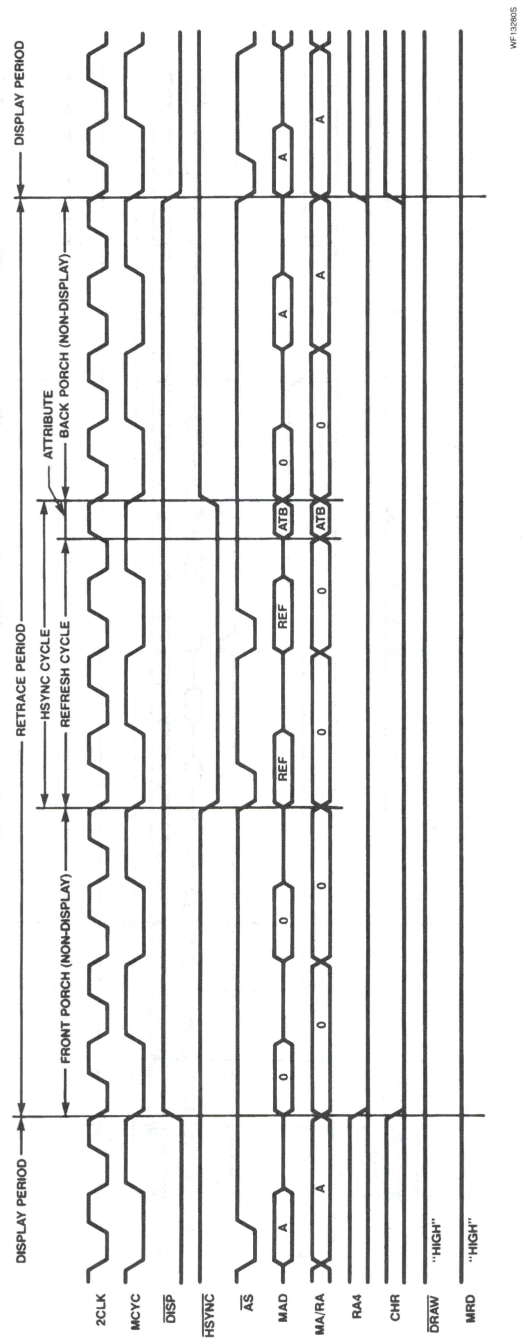

DISPLAY PERIOD
RETRACE PERIOD
DISPLAY PERIOD
FRONT PORCH (NON-DISPLAY)
HSYNC CYCLE
REFRESH CYCLE
ATTRIBUTE
BACK PORCH (NON-DISPLAY)
2CLK
MCYC
DISP
HSYNC
AS
MAD
MA/RA
RA4
CHR
DRAW
MRD
"HIGH"
"HIGH"
A
0
REF
ATB
WF13280S

Figure 38. DRAM Refresh Timing

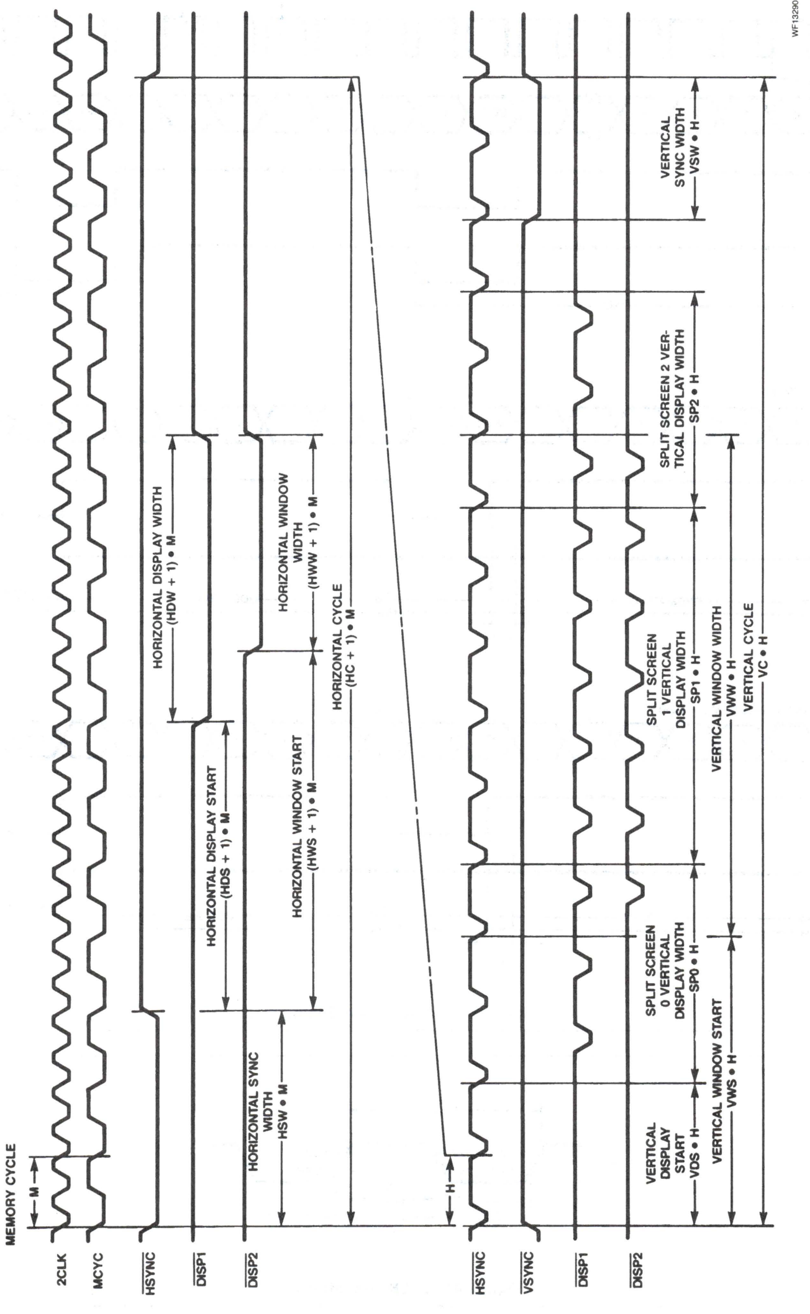

MEMORY CYCLE
M
2CLK
MCYC
HSYNC
DISP1
DISP2
HORIZONTAL SYNC WIDTH HSW • M
HORIZONTAL DISPLAY START (HDS + 1) • M
HORIZONTAL DISPLAY WIDTH (HDW + 1) • M
HORIZONTAL WINDOW START (HWS + 1) • M
HORIZONTAL WINDOW WIDTH (HWW + 1) • M
HORIZONTAL CYCLE (HC + 1) • M
H
HSYNC
VSYNC
DISP1
DISP2
VERTICAL DISPLAY START VDS • H
SPLIT SCREEN 0 VERTICAL DISPLAY WIDTH SP0 • H
SPLIT SCREEN 1 VERTICAL DISPLAY WIDTH SP1 • H
SPLIT SCREEN 2 VERTICAL DISPLAY WIDTH SP2 • H
VERTICAL SYNC WIDTH VSW • H
VERTICAL WINDOW START VWS • H
VERTICAL WINDOW WIDTH VWW • H
VERTICAL CYCLE VC • H
WF13290S

Figure 39. Display Screen Timing

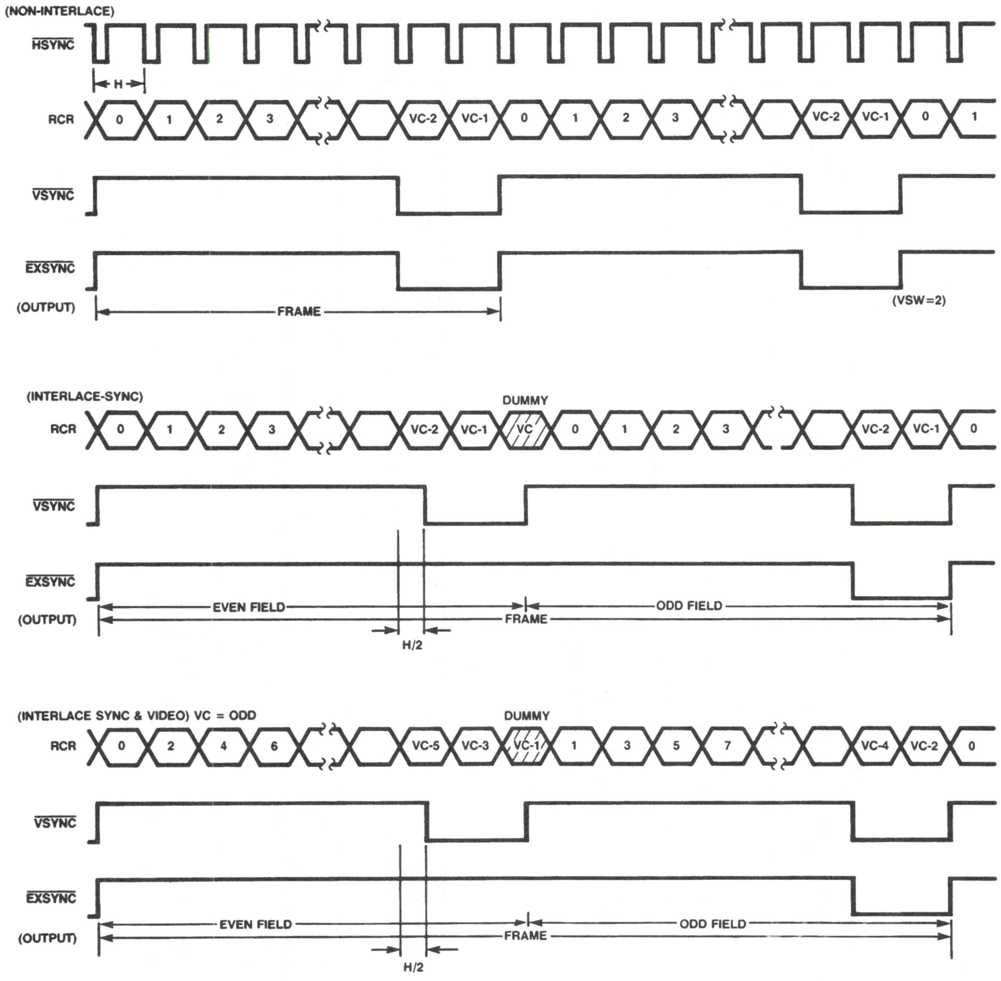

(NON-INTERLACE)
HSYNC
H
RCR
0
1
2
3
VC-2
VC-1
0
1
2
3
VC-2
VC-1
0
1
VSYNC
EXSYNC
(OUTPUT)
FRAME
(VSW=2)
(INTERLACE-SYNC)
DUMMY
RCR
0
1
2
3
VC-2
VC-1
VC
0
1
2
3
VC-2
VC-1
0
VSYNC
EXSYNC
(OUTPUT)
EVEN FIELD
ODD FIELD
FRAME
H/2
(INTERLACE SYNC & VIDEO) VC = ODD
DUMMY
RCR
0
2
4
6
VC-5
VC-3
VC-1
1
3
5
7
VC-4
VC-2
0
VSYNC
EXSYNC
(OUTPUT)
EVEN FIELD
ODD FIELD
FRAME
H/2
WF13300S

Figure 40. Raster Scan Timing

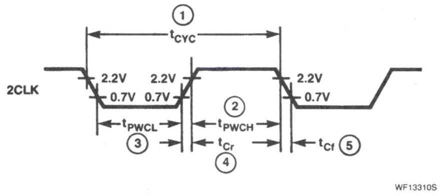
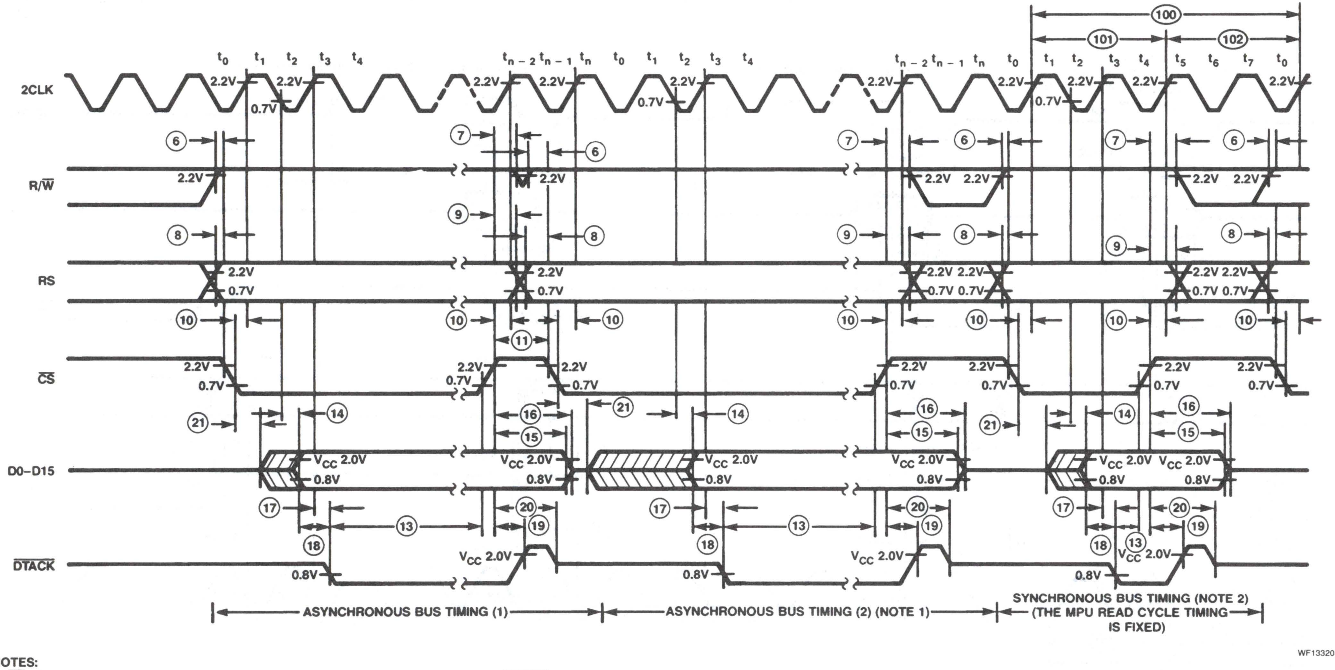
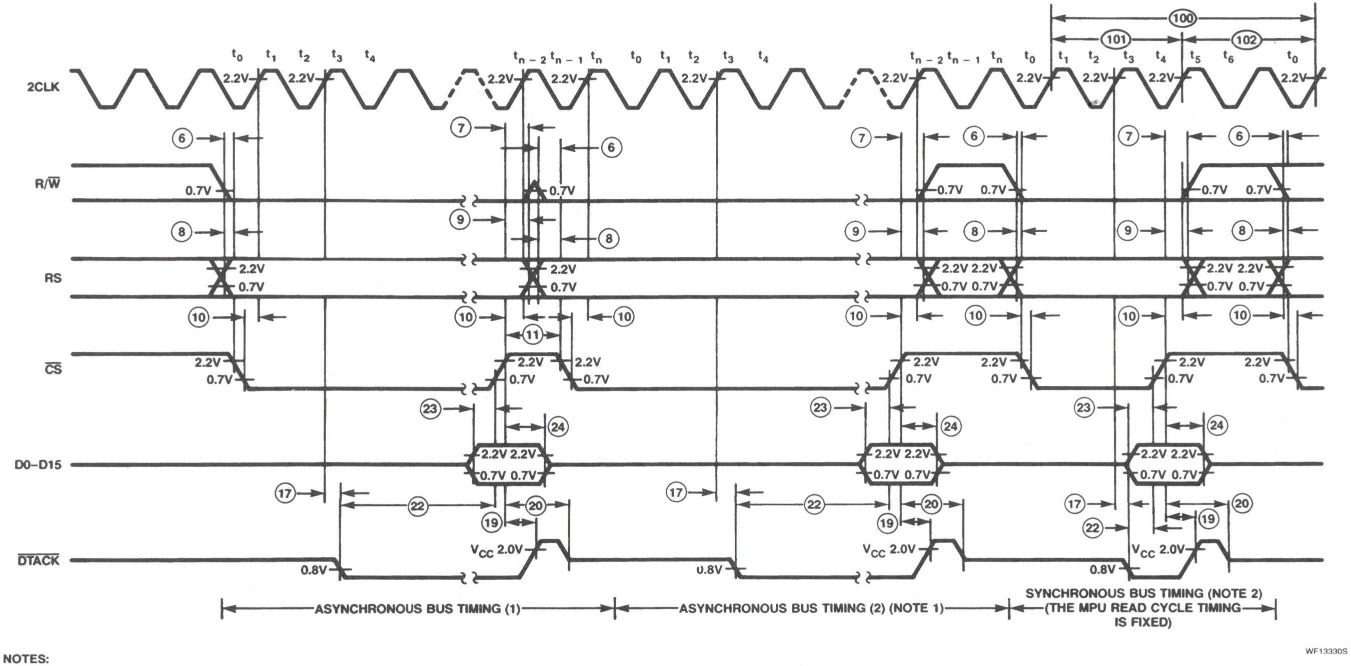

①
tCYC
2CLK
2.2V
2.2V
2.2V
0.7V
0.7V
0.7V
②
tPWCL
tPWCH
③
tCr
④
tCf
⑤
WF13310S

Figure 41. 2CLK Waveform


2CLK
R/W̄
RS
CS̄
D0–D15
DTACK̄
t_0 t_1 t_2 t_3 t_4
t_{n-2} t_{n-1} t_n
t_5 t_6 t_7
2.2V
0.7V
0.8V
V_{CC} 2.0V
⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ⑳ ㉑
(100) (101) (102)
ASYNCHRONOUS BUS TIMING (1)
ASYNCHRONOUS BUS TIMING (2) (NOTE 1)
SYNCHRONOUS BUS TIMING (NOTE 2) (THE MPU READ CYCLE TIMING IS FIXED)
WF13320S

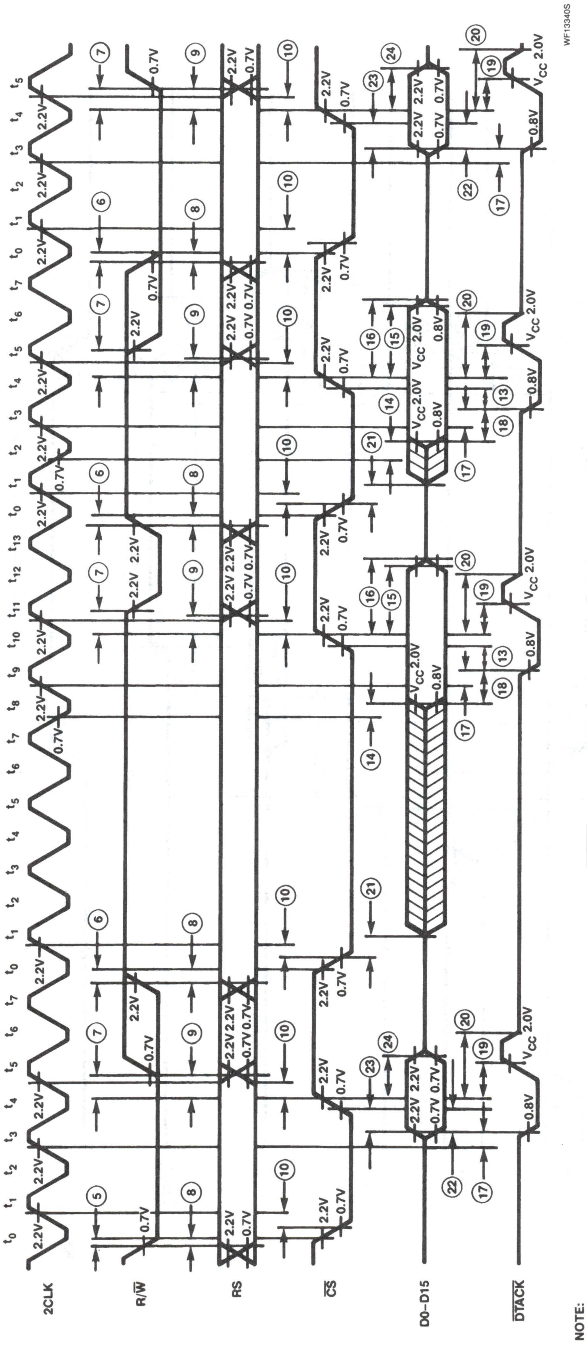
NOTES:
1. CS̄ "high" width must satisfy the specification ⑪. Unless satisfying the spec (102), DTACK̄ and read data responses to the succeeding cycle are delayed.
2. When the ACRTC is used with the synchronous bus timing, the specifications (100), (101) and (102) must be satisfied.

Figure 42. MPU Read Cycle Timing (MPU ← ACRTC)


2CLK
R/W̄
RS
CS̄
D0–D15
DTACK̄
t_0 t_1 t_2 t_3 t_4
t_{n-2} t_{n-1} t_n
t_5 t_6
2.2V
0.7V
0.8V
V_{CC} 2.0V
⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑰ ⑲ ⑳ ㉒ ㉓ ㉔
(100) (101) (102)
ASYNCHRONOUS BUS TIMING (1)
ASYNCHRONOUS BUS TIMING (2) (NOTE 1)
SYNCHRONOUS BUS TIMING (NOTE 2)
(THE MPU READ CYCLE TIMING IS FIXED)
WF13330S

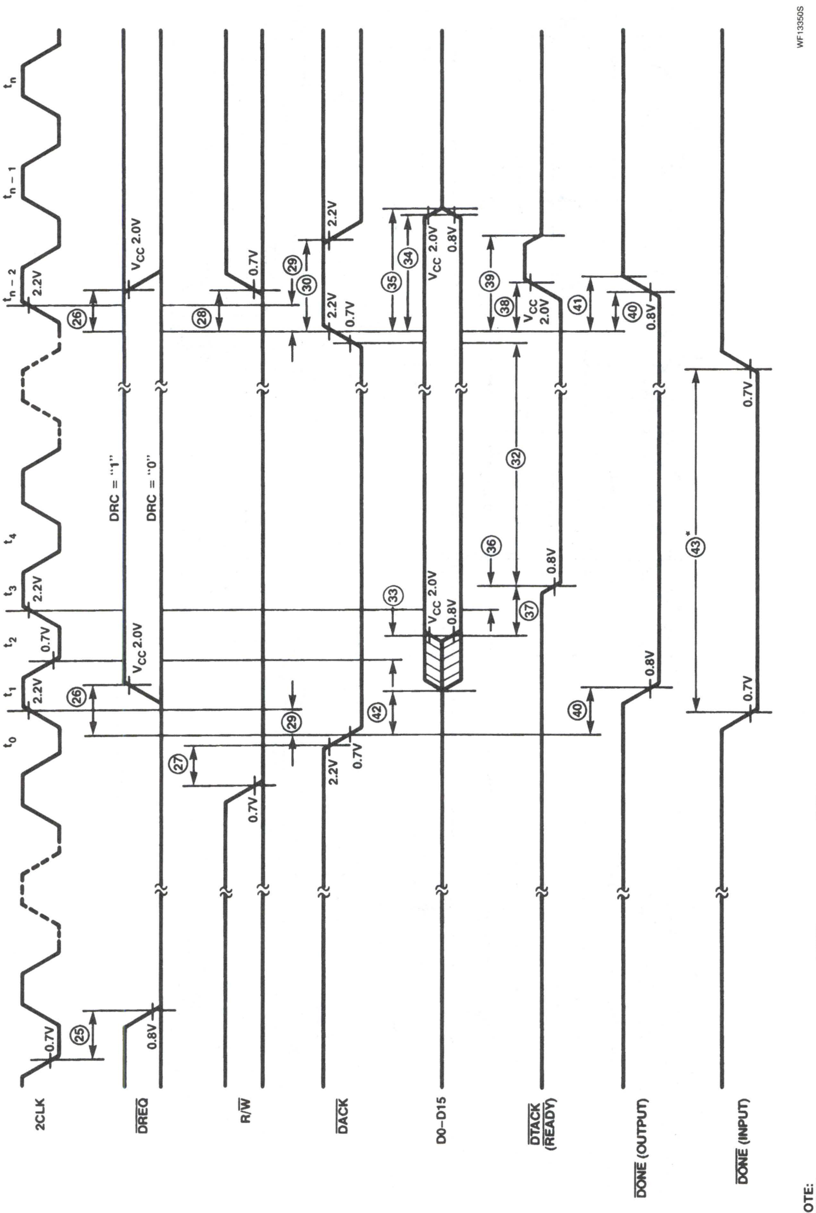
NOTES:
1. $\overline{CS}$ "high" width must satisfy the specification ⑪. Unless satisfying the spec (102), $\overline{DTACK}$ response to the succeeding cycle is delayed.
2. When the ACRTC is used with the synchronous bus timing, the specifications (100), (101) and (102) must be satisfied.

Figure 43. MPU Read/Write Cycle Timing (MPU → ACRTC)


2CLK
R/W̄
RS
CS̄
D0–D15
DTACK̄
t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13
2.2V
0.7V
0.8V
VCC 2.0V
⑤
⑥
⑦
⑧
⑨
⑩
⑬
⑭
⑮
⑯
⑰
⑱
⑲
⑳
㉑
㉒
㉓
㉔
WF13340S

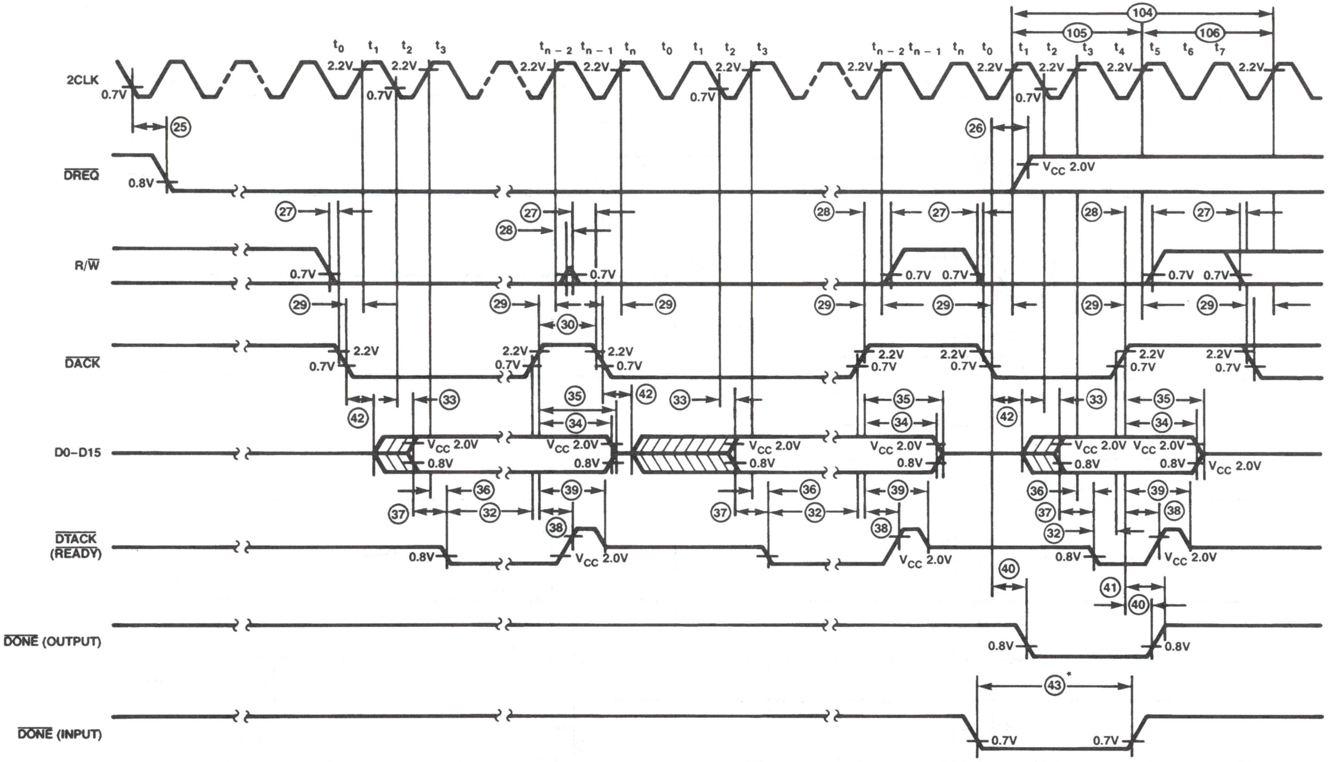
NOTE:
When the MPU read cycle immediately follows the MPU write cycle execution, DTACK̄ and the read data responses are delayed (by 3 cycles of 2CLK) even though the spec (102) is satisfied.

Figure 43. MPU Read/Write Cycle Timing (MPU → ACRTC) (Continued)


2CLK
DREQ
R/W
DACK
D0–D15
DTACK
(READY)
DONE (OUTPUT)
DONE (INPUT)
t0
t1
t2
t3
t4
tn - 2
tn - 1
tn
0.7V
2.2V
0.8V
VCC 2.0V
DRC = "1"
DRC = "0"
⑮
⑯
㉗
㉘
㉙
㉚
㉜
㉝
㉞
㉟
㊱
㊲
㊳
㊴
㊵
㊶
㊷
㊸*
WF13350S

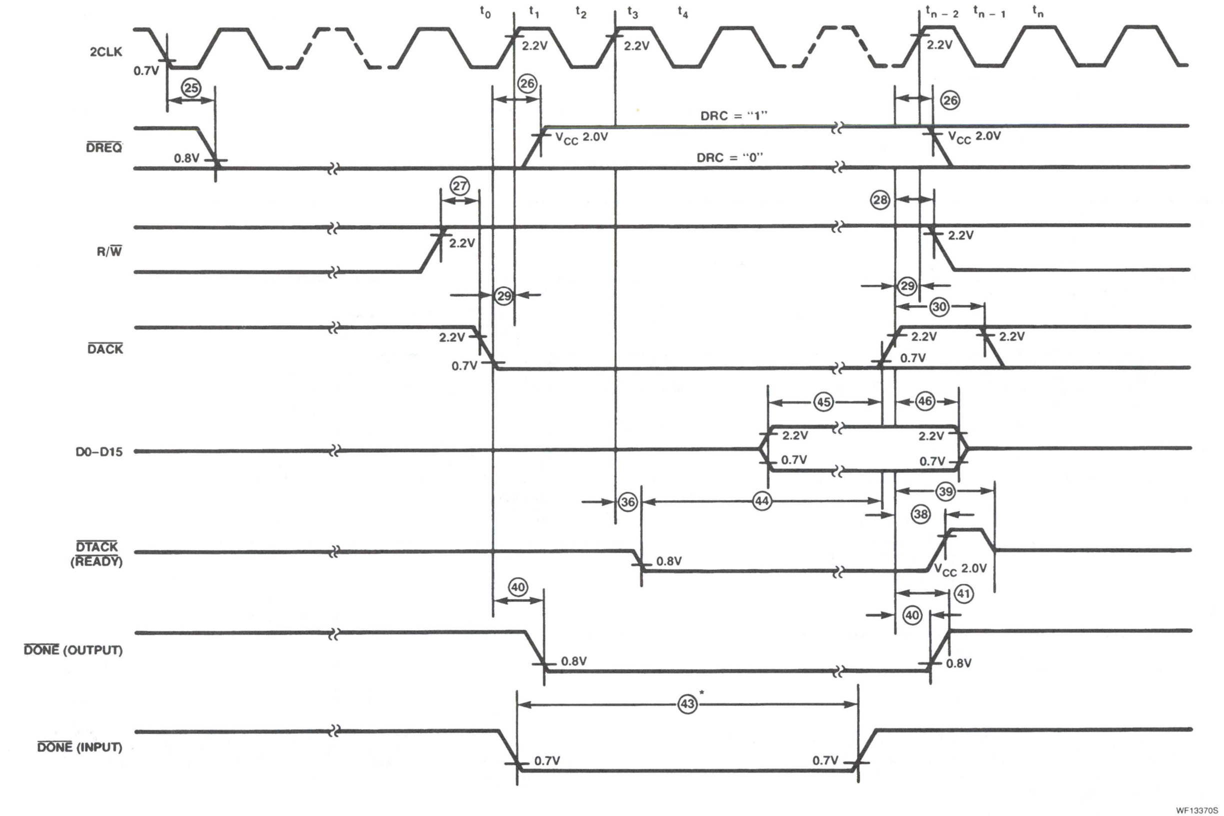
NOTE:
*DONE needs to be asserted "Low" while DACK remains "Low." DONE "Low" width must satisfy the spec ㊸

Figure 44. DMA Read Cycle Timing (Memory ← ACRTC)


2CLK
DREQ
R/W
DACK
D0–D15
DTACK (READY)
DONE (OUTPUT)
DONE (INPUT)
t_0 t_1 t_2 t_3
t_{n-2} t_{n-1} t_n
t_4 t_5 t_6 t_7
0.7V
0.8V
2.2V
V_{CC} 2.0V
⑳⑤
㉕
㉖
㉗
㉘
㉙
㉚
㉜
㉝
㉞
㉟
㊱
㊲
㊳
㊴
㊵
㊶
㊷
㊸*
(104)
(105)
(106)
WF13360S

NOTES:
1. *DONE needs to be asserted "Low" while DACK remains "Low."
2. DONE "Low" width must satisfy the spec ㊸.
3. DACK "high" width must satisfy the spec ㉚. Unless satisfying the spec (106) DTACK and the read data responses to the succeeding cycle are delayed. When the ACRTC is used with the synchronous bus timing, the specifications (104), (105) and (106) must be satisfied.

Figure 44. DMA Read Cycle Timing (Memory ← ACRTC): Burst Mode (Continued)

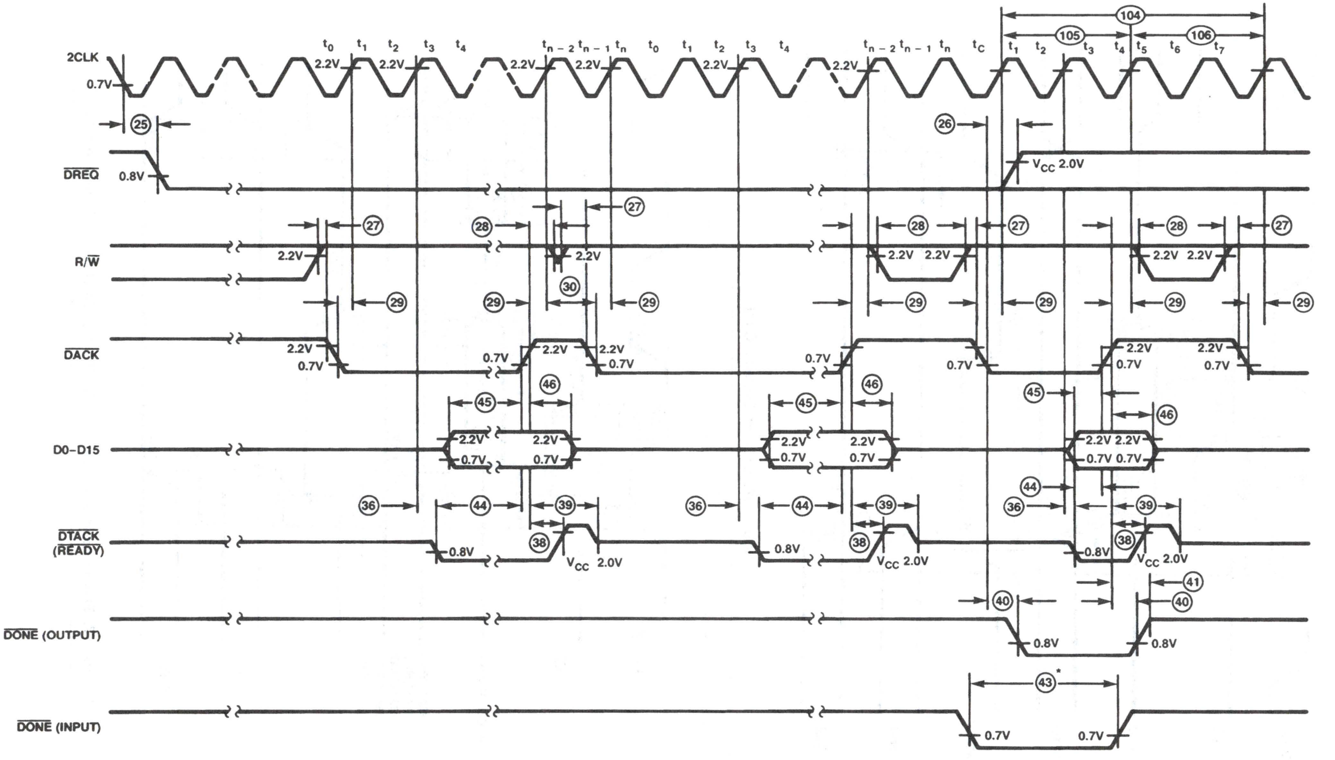

t0
t1
t2
t3
t4
tn - 2
tn - 1
tn
2CLK
0.7V
2.2V
2.2V
2.2V
㉕
㉖
㉖
DREQ
0.8V
VCC 2.0V
VCC 2.0V
DRC = "1"
DRC = "0"
㉗
㉘
R/W
2.2V
2.2V
㉙
㉙
㉚
DACK
2.2V
0.7V
2.2V
0.7V
2.2V
㊺
㊻
D0–D15
2.2V
0.7V
2.2V
0.7V
㊱
㊹
㊴
㊳
DTACK (READY)
0.8V
VCC 2.0V
㊵
㊶
㊵
DONE (OUTPUT)
0.8V
0.8V
㊸*
DONE (INPUT)
0.7V
0.7V
WF13370S

NOTE:
*DONE needs to be asserted "Low" while DACK remains "Low". DONE "Low" width must satisfy the spec ㊸.

Figure 45. DMA Write Cycle Timing (Memory → ACRTC)

Advanced CRT Controller (ACRTC)

SCC63484


2CLK
0.7V
2.2V
t0 t1 t2 t3 t4
tn-2 tn-1 tn
tC
t1 t2 t3 t4 t5 t6 t7
(104)
(105)
(106)
㉕
㉖
DREQ
0.8V
VCC 2.0V
R/W
㉗
㉘
㉙
㉚
DACK
D0-D15
㊺
㊻
DTACK (READY)
㊱
㊹
㊳
㊴
VCC 2.0V
0.8V
DONE (OUTPUT)
㊵
㊶
DONE (INPUT)
㊸*
WF13380S

NOTE:
1. *DONE needs to be asserted "Low" while DACK remains "low."
2. DONE "Low" width must satisfy the spec ㊸.
3. DACK "high" width must satisfy the spec ㉚. Unless satisfying the spec (106) DTACK responses to the succeeding cycle are delayed. When the ACRTC is used with the synchronous bus timing, the specifications (104), (105) and (106) must be satisfied.

Figure 45. DMA Write Cycle Timing (Memory → ACRTC): Burst Mode (Continued)

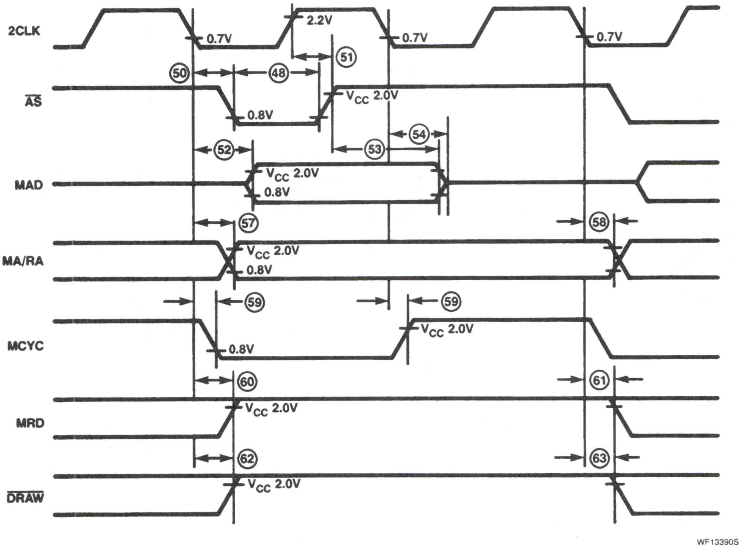

2CLK
0.7V
2.2V
0.7V
0.7V
(51)
(50)
(48)
AS
0.8V
VCC 2.0V
(54)
(52)
(53)
MAD
VCC 2.0V
0.8V
(57)
(58)
MA/RA
VCC 2.0V
0.8V
(59)
(59)
MCYC
0.8V
VCC 2.0V
(60)
(61)
MRD
VCC 2.0V
(62)
(63)
DRAW
VCC 2.0V
WF13390S

Figure 46. Screen Display Cycle Timing

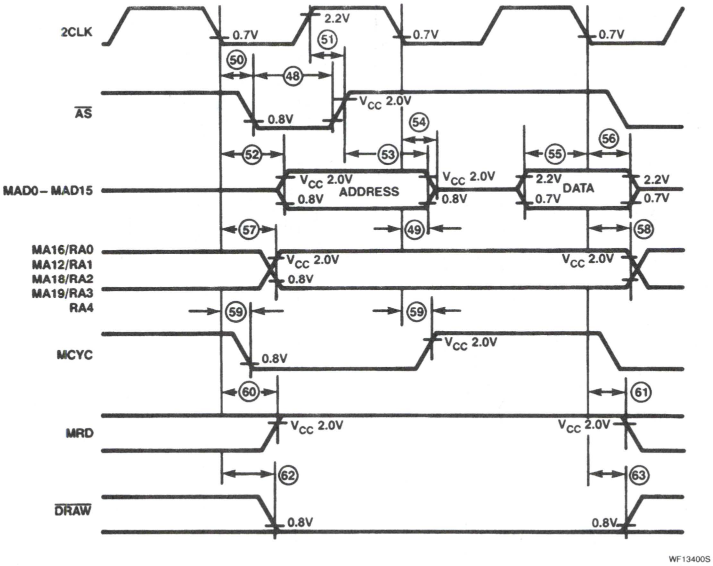

2CLK
0.7V
2.2V
0.7V
0.7V
(51)
(50)
(48)
AS
0.8V
VCC 2.0V
(54)
(56)
(52)
(53)
(55)
MAD0 – MAD15
VCC 2.0V
0.8V
ADDRESS
VCC 2.0V
0.8V
2.2V
0.7V
DATA
2.2V
0.7V
(57)
(49)
(58)
MA16/RA0
MA12/RA1
MA18/RA2
MA19/RA3
RA4
VCC 2.0V
0.8V
VCC 2.0V
(59)
(59)
MCYC
0.8V
VCC 2.0V
(60)
(61)
MRD
VCC 2.0V
VCC 2.0V
(62)
(63)
DRAW
0.8V
0.8V
WF13400S

Figure 47. Frame Memory Read Cycle Timing (ACRTC ← Frame Memory)

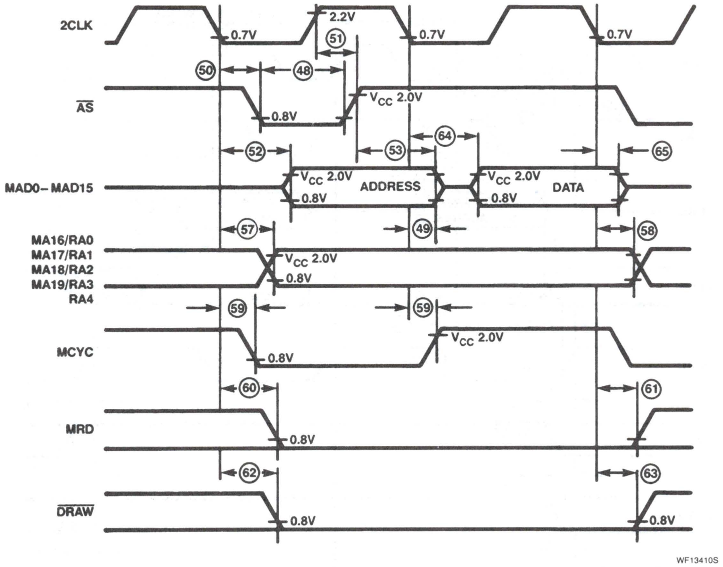
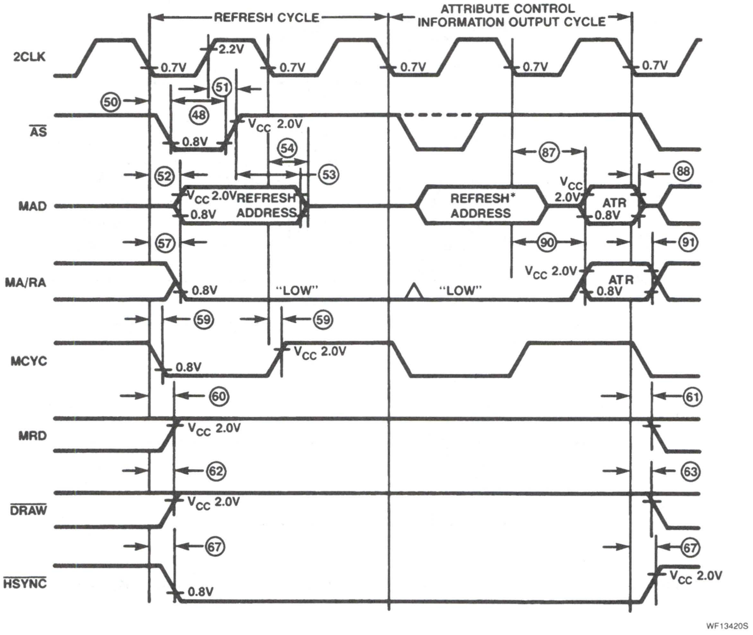

2CLK
0.7V
2.2V
(51)
0.7V
0.7V
(50)
(48)
AS
0.8V
VCC 2.0V
(64)
(52)
(53)
(65)
MAD0–MAD15
VCC 2.0V
0.8V
ADDRESS
VCC 2.0V
0.8V
DATA
(57)
(49)
(58)
MA16/RA0
MA17/RA1
MA18/RA2
MA19/RA3
RA4
VCC 2.0V
0.8V
(59)
(59)
MCYC
0.8V
VCC 2.0V
(60)
(61)
MRD
0.8V
(62)
(63)
DRAW
0.8V
0.8V
WF13410S

Figure 48. Frame Memory Write Cycle Timing (ACRTC → Frame Memory)


REFRESH CYCLE
ATTRIBUTE CONTROL INFORMATION OUTPUT CYCLE
2CLK
0.7V
2.2V
AS
MAD
MA/RA
MCYC
MRD
DRAW
HSYNC
(50)
(51)
(48)
(52)
(53)
(54)
(57)
(59)
(60)
(61)
(62)
(63)
(67)
(87)
(88)
(90)
(91)
0.8V
V_{CC} 2.0V
REFRESH ADDRESS
REFRESH* ADDRESS
ATR
"LOW"
WF13420S

NOTE:
*When $\overline{AS}$ is "High", a "0" output is given.

Figure 49. Frame Memory Refresh/Attribute Control Information Output Cycle Timing

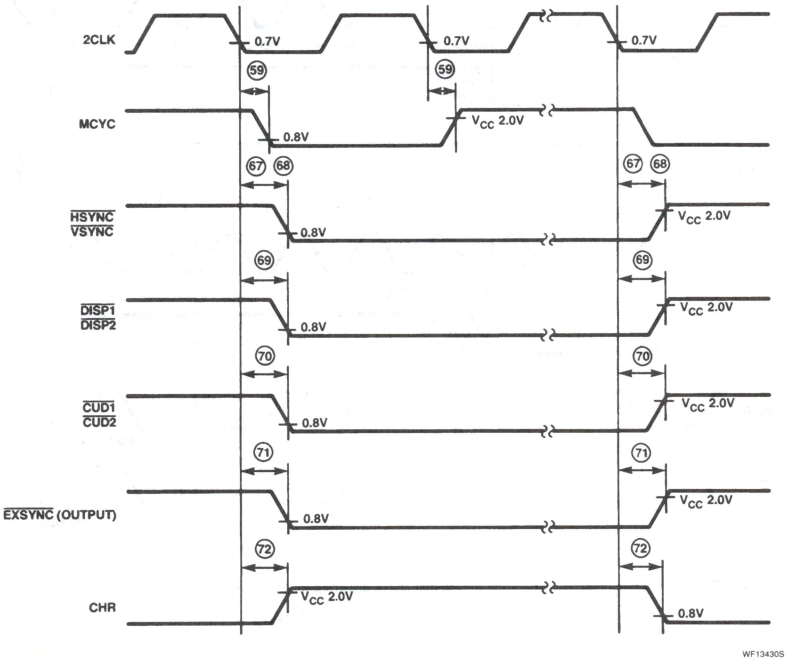
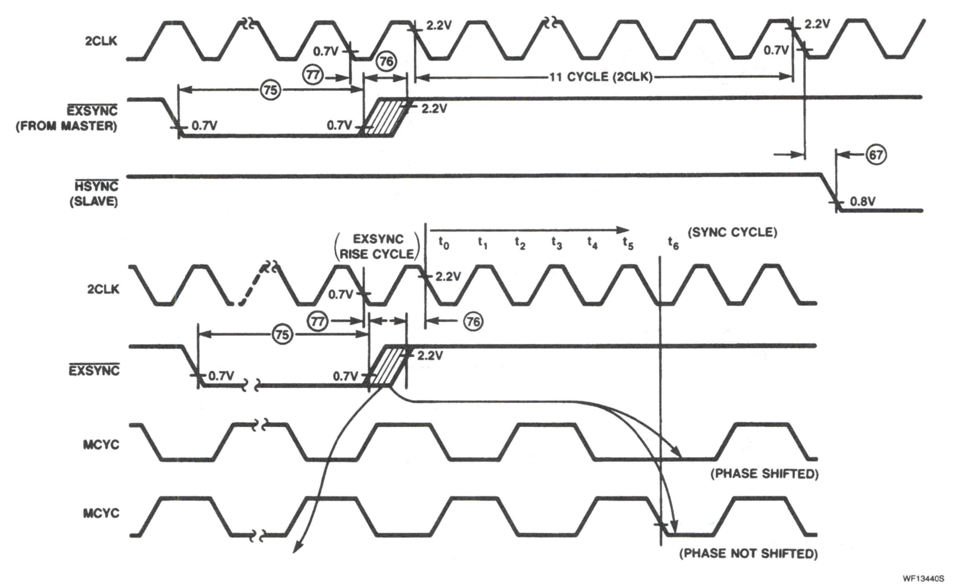
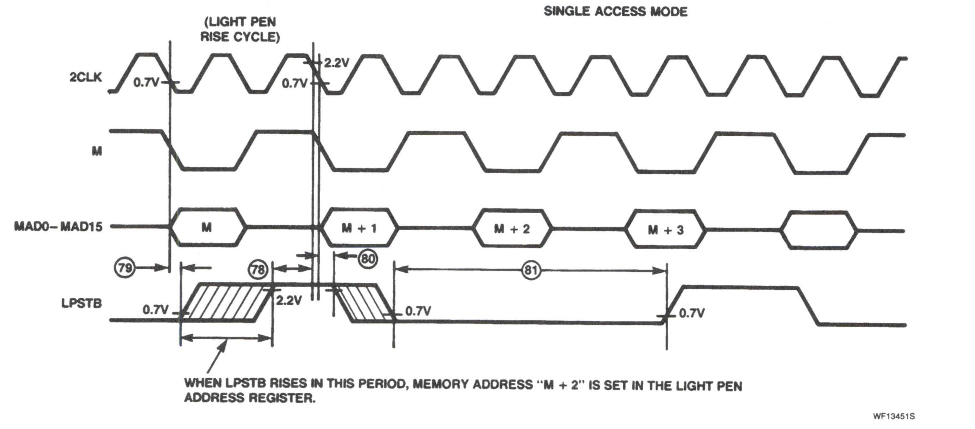

2CLK
0.7V
0.7V
0.7V
59
59
MCYC
0.8V
VCC 2.0V
67
68
67
68
HSYNC
VSYNC
0.8V
VCC 2.0V
69
69
DISP1
DISP2
0.8V
VCC 2.0V
70
70
CUD1
CUD2
0.8V
VCC 2.0V
71
71
EXSYNC (OUTPUT)
0.8V
VCC 2.0V
72
72
CHR
VCC 2.0V
0.8V
WF13430S

Figure 50. Display Control Signal Output Timing


2CLK
EXSYNC
(FROM MASTER)
HSYNC
(SLAVE)
2CLK
EXSYNC
MCYC
MCYC
2.2V
0.7V
0.8V
⑦⑤
⑦⑥
⑦⑦
⑥⑦
11 CYCLE (2CLK)
EXSYNC RISE CYCLE
t_0 t_1 t_2 t_3 t_4 t_5 t_6
(SYNC CYCLE)
(PHASE SHIFTED)
(PHASE NOT SHIFTED)
WF13440S

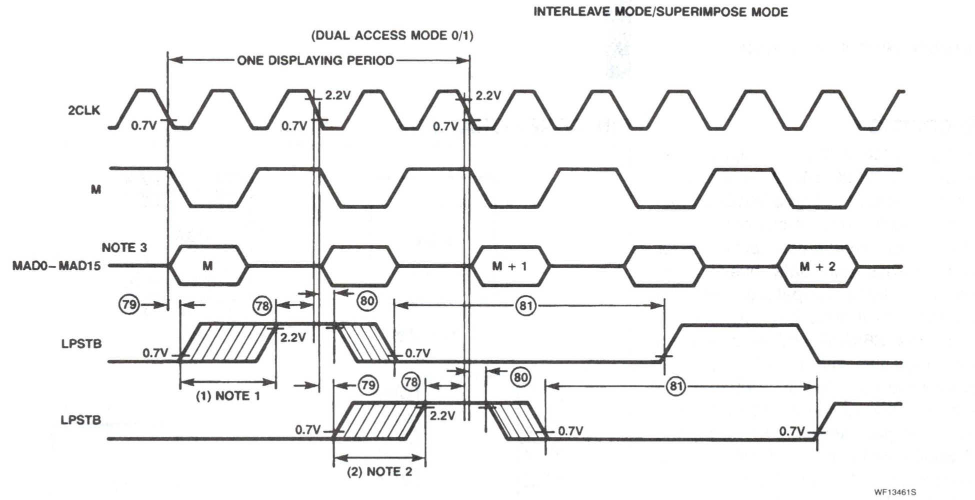
NOTE:
(When the leading edge of $\overline{\text{EXSYNC}}$ enters this period, ACRTC shifts the internal phase according to the above sequence.)

Figure 51. $\overline{\text{EXSYNC}}$ Input Timing


SINGLE ACCESS MODE
(LIGHT PEN
RISE CYCLE)
2CLK
M
MAD0-MAD15
LPSTB
0.7V
2.2V
M
M + 1
M + 2
M + 3
⑦⑧
⑦⑨
⑧⓪
⑧①
WHEN LPSTB RISES IN THIS PERIOD, MEMORY ADDRESS "M + 2" IS SET IN THE LIGHT PEN ADDRESS REGISTER.
WF13451S

NOTE:
1. Memory address set in light pen address register during H-SYNC cycle may not be proper.

Figure 52. LPSTB Input Timing


INTERLEAVE MODE/SUPERIMPOSE MODE
(DUAL ACCESS MODE 0/1)
ONE DISPLAYING PERIOD
2CLK
2.2V
0.7V
M
NOTE 3
MAD0–MAD15
M
M + 1
M + 2
(79)
(78)
(80)
(81)
LPSTB
(1) NOTE 1
(2) NOTE 2
WF13461S

NOTES:

1. When LPSTB rises in the period (1), memory address "M + 1" is set in the Light Pen Address register.
2. When LPSTB rises in the period (2), memory address "M + 2" is set in the "Light Pen Address register.
3. In the Interleave Mode, memory address "M", "M + 1", "M + 2" denote the display address. In the Superimpose Mode, memory address "M", "M + 1", "M + 2" denote the display address of the background screen.
4. Memory address set in light pen address register during H-SYNC cycle may not be proper.

Figure 52. LPSTB Input Timing (Continued)

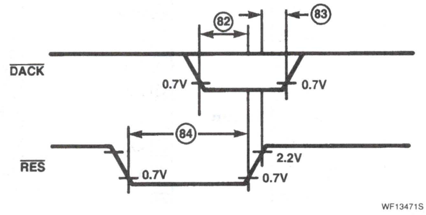

(82)
(83)
DACK
0.7V
(84)
RES
2.2V
0.7V
WF13471S

Figure 53. $\overline{\text{RES}}$ Input and $\overline{\text{DTACK}}$ Input Timing (System Reset and 16-Bit/8-Bit Selection)

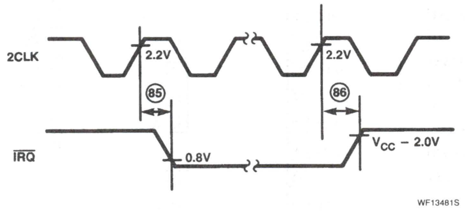

2CLK
2.2V
(85)
(86)
IRQ
0.8V
V_{CC} – 2.0V
WF13481S

Figure 54. $\overline{\text{IRQ}}$ Output Timing

Signetics

SCN68000
16-/32-Bit Microprocessor

Product Specification

Microprocessor Products

DESCRIPTION

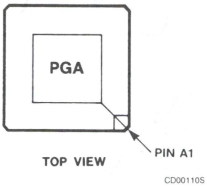
The SCN68000 is the first implementation of the S68000 16/32 bit microprocessor architecture. The SCN68000 has a 16-bit data bus and 24-bit address bus, while the full architecture provides for 32-bit address and data buses. It is completely code-compatible with the SCN68008 8-bit data bus implementation of the S68000 and is downward code-compatible with the SCN68010 virtual extension and the SCN68020 32-bit implementation of the architecture. Any user-mode programs written using the SCN68000 instruction set will run unchanged on the SCN68008, SCN68010, and 68020. This is possible because the user programming model is identical for all four processors and the instruction sets are proper sub-sets of the complete architecture.

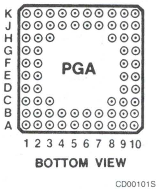
The resources available to the SCN68000 user consist of the following:

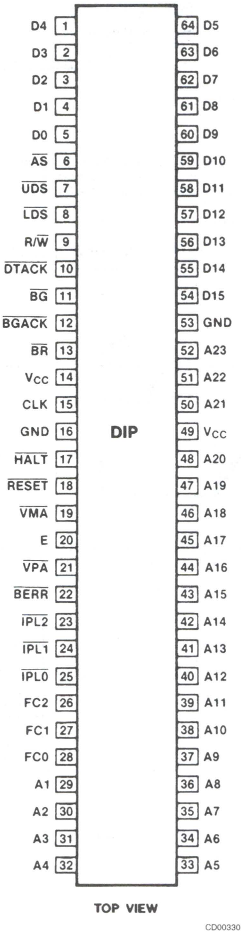
- 17 32-bit data and address registers
- 16 Mbyte direct addressing range
- 56 powerful instruction types
- Operations on five main data types
- Memory-mapped I/O
- 14 addressing modes

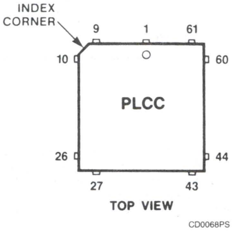
As shown in the programming model (Figure 1), the SCN68000 offers sixteen 32-bit registers and a 32-bit program counter. The first eight registers (D0 – D7) are used as data registers for byte (8-bit), word (16-bit), and long word (32-bit) operations. The second set of seven registers (A0 – A6) and the user stack pointer (USP) may be used as software stack pointers and base address registers. In addition, the registers may be used for word and long word operations. All of the 16 registers may be used as index registers.

PIN CONFIGURATIONS


PGA
PIN A1
TOP VIEW
CD00110S


K
J
H
G
F
E
D
C
B
A
PGA
1 2 3 4 5 6 7 8 9 10
BOTTOM VIEW
CD00101S


D4 1
D3 2
D2 3
D1 4
D0 5
AS 6
UDS 7
LDS 8
R/W 9
DTACK 10
BG 11
BGACK 12
BR 13
VCC 14
CLK 15
GND 16
HALT 17
RESET 18
VMA 19
E 20
VPA 21
BERR 22
IPL2 23
IPL1 24
IPL0 25
FC2 26
FC1 27
FC0 28
A1 29
A2 30
A3 31
A4 32
64 D5
63 D6
62 D7
61 D8
60 D9
59 D10
58 D11
57 D12
56 D13
55 D14
54 D15
53 GND
52 A23
51 A22
50 A21
49 VCC
48 A20
47 A19
46 A18
45 A17
44 A16
43 A15
42 A14
41 A13
40 A12
39 A11
38 A10
37 A9
36 A8
35 A7
34 A6
33 A5
DIP
TOP VIEW
CD00330S


INDEX CORNER
9
1
61
10
60
PLCC
26
44
27
43
TOP VIEW
CD0068PS

PLCC	Function	PLCC	Function	PGA	Function	PGA	Function
1	D4	35	A4	A1	NC	F1	$\overline{HALT}$
2	D3	36	A5	A2	$\overline{AS}$	F2	$\overline{RESET}$
3	D2	37	A6	A3	D1	F9	A18
4	D1	38	A7	A4	D2	F10	A19
5	D0	39	A8	A5	D4	G1	$\overline{VMA}$
6	AS	40	A9	A6	D5	G2	$\overline{VPA}$
7	$\overline{UDS}$	41	A10	A7	D7	G9	A15
8	$\overline{LDS}$	42	A11	A8	D8	G10	A17
9	R/$\overline{W}$	43	A12	A9	D10	H1	E
10	$\overline{DTACK}$	44	A13	A10	D12	H2	$\overline{IPL2}$
11	$\overline{BG}$	45	A14	B1	$\overline{DTACK}$	H3	$\overline{IPL1}$
12	$\overline{BGACK}$	46	A15	B2	$\overline{LDS}$	H8	A13
13	$\overline{BR}$	47	A16	B3	$\overline{UDS}$	H9	A12
14	V_{CC}	48	A17	B4	D0	H10	A16
15	CLK	49	A18	B5	D3	J1	$\overline{BERR}$
16	GND	50	A19	B6	D6	J2	$\overline{IPL0}$
17	GND	51	A20	B7	D9	J3	FC1
18	NC	52	V_{CC}	B8	D11	J4	NC
19	$\overline{HALT}$	53	A21	B9	D14	J5	A2
20	$\overline{RESET}$	54	A22	B10	D15	J6	A5
21	$\overline{VMA}$	55	A23	C1	$\overline{BGACK}$	J7	A8
22	E	56	GND	C2	$\overline{BG}$	J8	A10
23	$\overline{VPA}$	57	GND	C3	R/$\overline{W}$	J9	A11
24	$\overline{BERR}$	58	D15	C8	D13	J10	A14
25	$\overline{IPL2}$	59	D14	C9	A23	K1	NC
26	$\overline{IPL1}$	60	D13	C10	A22	K2	FC2
27	$\overline{IPL0}$	61	D12	D1	$\overline{BR}$	K3	FC0
28	FC2	62	D11	D2	V_{CC}	K4	A1
29	FC1	63	D10	D9	GND	K5	A3
30	FC0	64	D9	D10	A21	K6	A4
31	NC	65	D8	E1	CLK	K7	A6
32	A1	66	D7	E2	GND	K8	A7
33	A2	67	D6	E9	V_{CC}	K9	A9
34	A3	68	D5	E10	A20	K10	NC

16-/32-Bit Microprocessor SCN68000

ORDERING INFORMATION

PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to 70°C		
	8MHz	10MHz	12.5MHz
64-Pin Ceramic DIP	SCN68000C8I64	SCN68000CAI64	SCN68000CBI64
64-Pin Plastic DIP	SCN68000C8N64	SCN68000CAN64	SCN68000CBN64
68-Pin Plastic LCC	SCN68000C8A68	SCN68000CAA68	SCN68000CBA68
68-Pin PGA	SCN68000C8P68	SCN68000CP68	SCN68000CBP68

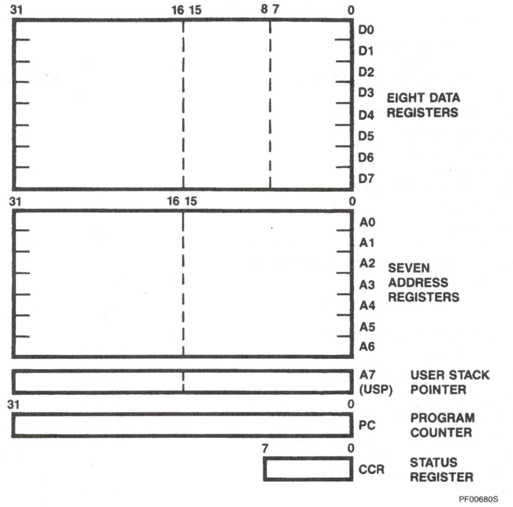

31
16 15
8 7
0
D0
D1
D2
D3
D4
D5
D6
D7
EIGHT DATA REGISTERS
31
16 15
0
A0
A1
A2
A3
A4
A5
A6
SEVEN ADDRESS REGISTERS
A7 (USP)
USER STACK POINTER
31
0
PC
PROGRAM COUNTER
7
0
CCR
STATUS REGISTER
PF00680S

Figure 1. User Programming Model

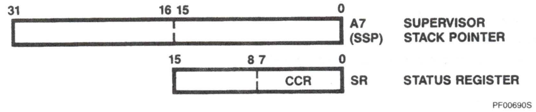

31
16 15
0
A7 (SSP)
SUPERVISOR STACK POINTER
15
8 7
0
CCR
SR
STATUS REGISTER
PF00690S

Figure 2. Supervisor Programming Model Supplement

In supervisor mode, the upper byte of the status register and the supervisor stack pointer (SSP) are also available to the programmer. These registers are shown in Figure 2.

The status register (Figure 3) contains the interrupt mask (eight levels available) as well as the condition codes: extend (X), negative (N), zero (Z), overflow (V), and carry (C). Additional status bits indicate that the processor is in a trace (T) mode and in a supervisor (S) or user state.

Data Types and Addressing Modes

Five basic data types are supported. These data types are:

- Bits
- BCD digits (4 bits)
- Bytes (8 bits)
- Words (16 bits)
- Long words (32 bits)

In addition, operations on other data types such as memory addresses, status word data, etc., are provided in the instruction set.

The 14 address modes, shown in Table 1, include six basic types:

- Register direct
- Register indirect
- Absolute
- Program counter relative
- Immediate
- Implied

Included in the register indirect addressing modes is the capability to do postincrementing, predecrementing, offsetting, and indexing. The program counter relative mode can also be modified via indexing and offsetting.

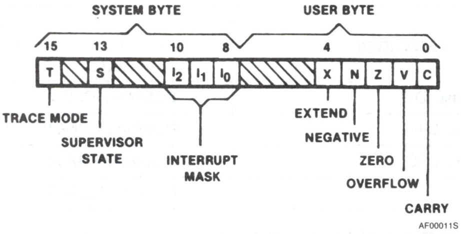


Figure 3. Status Register

Table 1. Addressing Modes

ADDRESSING MODES	SYNTAX
Register direct addressing	
Data register direct	Dn
Address register direct	An
Absolute data addressing	
Absolute short	xxx.W
Absolute long	xxx.L
Program counter relative addressing	
Relative with offset	d_{16}(PC)
Relative with index and offset	d_8(PC,Xn)
Register indirect addressing	
Register indirect	(An)
Postincrement register indirect	(An) +
Predecrement register indirect	−(An)
Register indirect with offset	d_{16}(An)
Indexed register indirect with offset	d_8(An,Xn)
Immediate data addressing	
Immediate	#xxx
Quick immediate	#1 − #8
Implied addressing	
Implied register	SR/USP/SP/PC

NOTES:
Dn = Data register
An = Address register
Xn = Address or data register used as index register
SR = Status register
PC = Program counter
SP = Stack pointer
USP = User stack pointer
( ) = Effective Address
d_8 = 8-bit offset (displacement)
d_{16} = 16-bit offset (displacement)
#xxx = Immediate data

16-/32-Bit Microprocessor SCN68000

Instruction Set Overview

The SCN68000 instruction set is shown in Table 2. Some additional instructions are variations, or subsets, of these and they appear in Table 3. Special emphasis has been given to the instruction set's support of structured high-level languages to facilitate ease of programming. Each instruction, with few exceptions, operates on bytes, words, and long words and most instructions can use any of the 14 addressing modes. Combining instruction types, data types, and addressing modes, over 1000 useful instructions are provided. These instructions include signed and unsigned, multiply and divide, ''quick'' arithmetic operations, BCD arithmetic, and expanded operations (through traps).

Table 2. Instruction Set Summary

MNEMONIC	DESCRIPTION
ABCD	Add decimal with extend
ADD	Add
AND	Logical AND
ASL	Arithmetic shift left
ASR	Arithmetic shift right
B_{CC}	Branch conditionally
BCHG	Bit test and change
BCLR	Bit test and clear
BRA	Branch always
BSET	Bit test and set
BSR	Branch to subroutine
BTST	Bit test
CHK	Check register against bounds
CLR	Clear operand
CMP	Compare
DB_{CC}	Test condition, decrement and branch
DIVS	Signed divide
DIVU	Unsigned divide
EOR	Exclusive OR
EXG	Exchange registers
EXT	Sign extend
JMP	Jump
JSR	Jump to subroutine
LEA	Load effective address
LINK	Link stack
LSL	Logical shift left
LSR	Logical shift right
MOVE	Move source to destination
MULS	Signed multiply
MULU	Unsigned multiply
NBCD	Negate decimal with extend
NEG	Negate
NOP	No operation
NOT	One's complement
OR	Logical OR
PEA	Push effective address
RESET	Reset external devices
ROL	Rotate left without extend
ROR	Rotate right without extend
ROXL	Rotate left with extend
ROXR	Rotate right with extend
RTE	Return from exception
RTR	Return and restore
RTS	Return from subroutine
SBCD	Subtract decimal with extend
S_{CC}	Set conditional
STOP	Stop
SUB	Subtract
SWAP	Swap data register halves
TAS	Test and set operand
TRAP	Trap
TRAPV	Trap on overflow
TST	Test
UNLK	Unlink

Table 3. Variations of Instruction Types

INSTRUCTION TYPE	VARIATION	DESCRIPTION
ADD	ADD ADDA ADDQ ADDI ADDX	Add Add address Add quick Add immediate Add with extend
AND	AND ANDI ANDI to CCR ANDI to SR	Logical AND AND immediate AND immediate to condition codes AND immediate to status register
CMP	CMP CMPA CMPM CMPI	Compare Compare address Compare memory Compare immediate
EOR	EOR EORI EORI to CCR EORI to SR	Exclusive OR Exclusive OR immediate Exclusive OR immediate to condition codes Exclusive OR immediate to status register
MOVE	MOVE MOVEA MOVEM MOVEP MOVEQ MOVE from SR MOVE to SR MOVE to CCR MOVE USP	Move source to destination Move address Move multiple registers Move peripheral data Move quick Move from status register Move to status register Move to condition codes Move user stack pointer
NEG	NEG NEGX	Negate Negate with extend
OR	OR ORI ORI to CCR ORI to SR	Logical OR OR immediate OR immediate to condition codes OR immediate to status register
SUB	SUB SUBA SUBI SUBQ SUBX	Subtract Subtract address Subtract immediate Subtract quick Subtract with extend

DATA ORGANIZATION AND ADDRESSING CAPABILITIES

This section contains a description of the registers and the data organization of the SCN68000.

Operand Size

Operand sizes are defined as follows: a byte equals 8 bits, a word equals 16 bits, and a long word equals 32 bits. The operand size for each instruction is either explicitly encoded in the instruction or implicitly defined by the instruction operation. Implicit instructions support some subset of all three sizes.

Data Organization in Registers

The eight data registers support data operands of 1, 8, 16, or 32 bits. The seven address registers together with the stack pointers support address operands of 32 bits.

Data Registers

Each data register is 32 bits wide. Byte operands occupy the low-order 8 bits, word operands the low-order 16 bits, and long-word operands the entire 32 bits. The least significant bit is addressed as bit zero; the most significant bit is addressed as 31.

When a data register is used as either a source or destination operand, only the appropriate low-order portion is changed; the remaining high-order portion is neither used nor changed.

Address Registers

Each address register and the stack pointer is 32 bits wide and holds a full 32-bit address. Address registers do not support the sized operands. Therefore, when an address register is used as a source operand, either the low-order word or the entire long-word operand is used depending on the operation size. When an address register is used as the destination operand, the entire register is affected regardless of the operation size. If the operation size is word, any other operands are sign extended to 32 bits before the operation is performed.

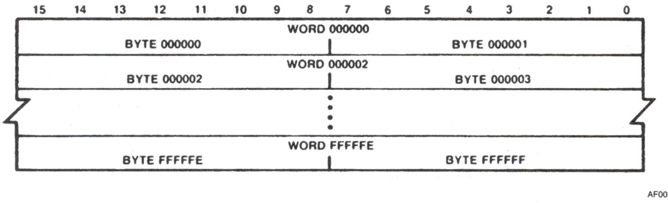


Figure 4. Word Organization in Memory

Data Organization in Memory

Bytes are individually addressable with the high-order byte having an even address the same as the word, as shown in Figure 4. The low-order byte has an odd address that is one count higher than the word address. Instructions and multibyte data are accessed only on word (even byte) boundaries. If a long word datum is located at address n (n even), then the second word of that datum is located at address n + 2.

The data types supported by the SCN68000 are: bit data, integer data of 8, 16, or 32 bits, 32-bit addresses and binary-coded decimal data. Each of these data types is put in memory, as shown in Figure 5. The numbers indicate the order in which the data would be accessed from the processor.

Addressing

Instructions for the SCN68000 contain two kinds of information: the type of function to be performed and the location of the operand(s) on which to perform that function. The methods used to locate (address) the operand(s) are explained in the following paragraphs.

Instructions specify an operand location in one of three ways:

Register specification – the number of the register is given in the register field of their instruction.

Effective address – use of the different effective addressing modes.

Implicit reference – the definition of certain instructions implies the use of specific registers.

Instruction Format

Instructions are from one to five words in length as shown in Figure 6. The length of the instruction and the operation to be performed is specified by the first word of the instruction which is called the operation word. The remaining words further specify the operands. These words are either immediate operands or extensions to the effective address mode specified in the operation word.

Program/Data References

The SCN68000 separates memory references into two classes: program references and data references. Program references, as the name implies, are references to that section of memory that contains the program being executed. Data references refer to that section of memory that contains data. Operand reads are from the data space except in the case of the program counter relative addressing mode. All operand writes are to the data space.

Register Specification

The register field within an instruction specifies the register to be used. Other fields within the instruction specify whether the register selected is an address or data register and how the register is to be used.

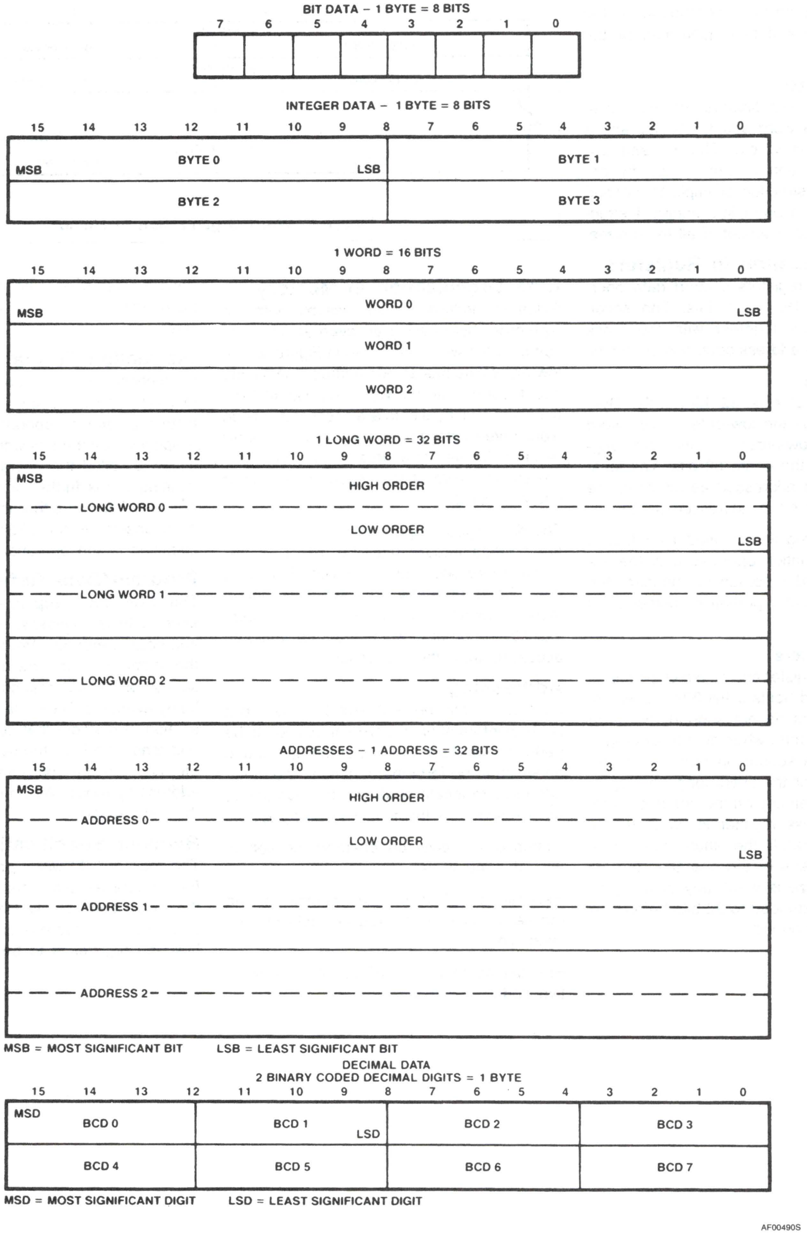
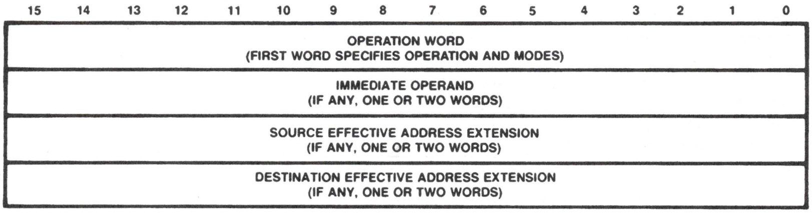
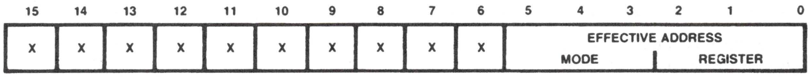

BIT DATA – 1 BYTE = 8 BITS
7 6 5 4 3 2 1 0
INTEGER DATA – 1 BYTE = 8 BITS
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
BYTE 0
LSB
BYTE 1
BYTE 2
BYTE 3
1 WORD = 16 BITS
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
WORD 0
LSB
WORD 1
WORD 2
1 LONG WORD = 32 BITS
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
HIGH ORDER
LONG WORD 0
LOW ORDER
LSB
LONG WORD 1
LONG WORD 2
ADDRESSES – 1 ADDRESS = 32 BITS
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
HIGH ORDER
ADDRESS 0
LOW ORDER
LSB
ADDRESS 1
ADDRESS 2
MSB = MOST SIGNIFICANT BIT
LSB = LEAST SIGNIFICANT BIT
DECIMAL DATA
2 BINARY CODED DECIMAL DIGITS = 1 BYTE
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSD
BCD 0
BCD 1
LSD
BCD 2
BCD 3
BCD 4
BCD 5
BCD 6
BCD 7
MSD = MOST SIGNIFICANT DIGIT
LSD = LEAST SIGNIFICANT DIGIT
AF00490S

Figure 5. Memory Data Organization


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
OPERATION WORD
(FIRST WORD SPECIFIES OPERATION AND MODES)
IMMEDIATE OPERAND
(IF ANY, ONE OR TWO WORDS)
SOURCE EFFECTIVE ADDRESS EXTENSION
(IF ANY, ONE OR TWO WORDS)
DESTINATION EFFECTIVE ADDRESS EXTENSION
(IF ANY, ONE OR TWO WORDS)

AF00510S

Figure 6. Instruction Operation Word General Format



AF00520S

Figure 7. Single-Effective-Address Instruction Operation Word

Effective Address

Most instructions specify the location of an operand by using the effective address field in the operation word. For example, Figure 7 shows the general format of the single-effective-address instruction operation word. The effective address is composed of two 3-bit fields: the mode field and the register field. The value in the mode field selects the different address modes. The register field contains the number of a register.

The effective address field may require additional information to fully specify the operand. This additional information, called the effective address extension, is contained in the following word or words and is considered part of the instruction, as shown in Figure 6. The effective address modes are grouped into three categories: register direct, memory addressing, and special.

Register Direct Modes

These effective addressing modes specify that the operand is in one of 16 multifunction registers.

Data Register Direct — The operand is in the data register specified by the effective address register field.

Address Register Direct — The operand is in the address register specified by the effective address register field.

Memory Address Modes

These effective addressing modes specify that the operand is in memory and provide the specific address of the operand.

Address Register Indirect — The address of the operand is in the address register specified by the register field. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Address Register Indirect with Postincrement — The address of the operand is in the address register specified by the register field. After the operand address is used, it is incremented by one, two, or four depending on whether the size of the operand is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is incremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

Address Register Indirect with Predecrement — The address of the operand is in the address register specified by the register field. Before the operand address is used, it is decremented by one, two, or four depending upon whether the operand size is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is decremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

Address Register Indirect with Displacement — This addressing mode requires one word of extension. The address of the operand is the sum of the address in the address register and the sign-extended 16-bit displacement integer in the extension word. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Address Register Indirect with Index — This addressing mode requires one word of extension. The address of the operand is the sum of the address in the address register, the sign-extended displacement integer in the low order eight bits of the extension word, and the contents of the index register. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Special Address Modes

The special address modes use the effective address register field to specify the special addressing mode instead of a register number.

Absolute Short Address — This addressing mode requires one word of extension. The address of the operand is the extension word. The 16-bit address is sign extended before it is used. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Absolute Long Address — This addressing mode requires two words of extension. The address of the operand is developed by the concatenation of the extension words. The high order part of the address is the first extension word; the low order part of the address is the second extension word. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Program Counter with Displacement — This addressing mode requires one word of extension. The address of the operand is the sum of the address in the program counter and the sign-extended 16-bit displacement integer in the word. The value in the program counter is the address of the extension word. The reference is classified as a program reference.

Program Counter with Index — This addressing mode requires one word of extension. The address is the sum of the address in the program counter, the sign-extended displacement integer in the lower eight bits of the extension word, and the contents of the index register. The value in the program counter is the address of the extension word. This reference is classified as a program reference.

Immediate Data — This addressing mode requires either one or two words of extension depending on the size of the operation.

Byte operation – operand is low-order byte of extension word.

Word operation – operand is extension word.

Long-word operation – operand is in the two extension words, high-order 16 bits are in the first extension word, low-order 16 bits are in the second extension word.

Implicit Reference — Some instructions make implicit reference to the program counter (PC), the system stack pointer (SP), the supervisor stack pointer (SSP), the user stack pointer (USP), or the status register (SR). A selected set of instructions may reference the status register by means of the effective address field. These are:

ANDI to CCR
ANDI to SR
EORI to CCR
EORI to SR
ORI to CCR
ORI to SR
MOVE to CCR
MOVE to SR
MOVE from SR

Effective Address Encoding Summary

Table 4 is a summary of the effective addressing modes discussed in the previous paragraphs.

System Stack

The system stack is used implicitly by many instructions; user stacks and queues may be created and maintained through the addressing modes. Address register seven (A7) is the system stack pointer (SP). The system stack pointer is either the supervisor stack pointer (SSP) or the user stack pointer (USP), depending on the state of the S bit in the status register. If the S bit indicates supervisor state, SSP is the active system stack pointer and the USP cannot be referenced as an address register. If the S bit indicates user state, the USP is the active system stack pointer and the SSP cannot be referenced. Each system stack fills from high memory to low memory.

Table 4. Effective Address Encoding Summary

ADDRESSING MODE	MODE	REGISTER
Data register direct	000	Register number
Address register direct	001	Register number
Address register indirect	010	Register number
Address register indirect with postincrement	011	Register number
Address register indirect with predecrement	100	Register number
Address register indirect with displacement	101	Register number
Address register indirect with index	110	Register number
Absolute short	111	000
Absolute long	111	001
Program counter with displacement	111	010
Program counter with index	111	011
Immediate	111	100

Table 5. Data Movement Operations

INSTRUCTION	OPERAND SIZE	OPERATION
EXG	32	Rx ↔ Ry
LEA	32	EA → An
LINK	–	AN → –(SP) SP → An SP + displacement → SP
MOVE	8, 16, 32	s → (EA)d
MOVEM	16, 32	(EA) → An, Dn An, Dn → EA
MOVEP	16, 32	(EA) → Dn Dn → (EA)
MOVEQ	8	#xxx → Dn
PEA	32	EA → –(SP)
SWAP	32	Dn[31:16] ↔ Dn[15:0]
UNLK	–	An → SP (SP) + → An

NOTES:
s = source
d = destination
[ ] = bit number
–( ) = indirect with predecrement
( ) + = indirect with postdecrement
= immediate data

INSTRUCTION SET SUMMARY

This section contains an overview of the form and structure of the SCN68000 instruction set. The instructions form a set of tools that include all the machine functions to perform the following operations:

Data movement
Integer arithmetic
Logical
Shift and rotate
Bit manipulation
Binary coded decimal
Program control
System control

The complete range of instruction capabilities combined with the flexible addressing modes described previously provide a very flexible base for program development.

Data Movement Operations

The basic method of data acquisition (transfer and storage) is provided by the move (MOVE) instruction. The move instruction and the effective addressing modes allow both address and data manipulation. Data move instructions allow byte, word, and long-word operands to be transferred from memory to memory, memory to register, register to memory, and register to register. Address move instructions allow word and long-word operand transfers and ensure that only legal address manipulations are executed. In addition to the general move instruction there are several special data movement instructions: move multiple registers (MOVEM), move peripheral data (MOVEP), exchange registers (EXG), load effective address (LEA), push effective address (PEA), link stack (LINK), unlink stack (UNLK), and move quick (MOVEQ). Table 5 is a summary of the data movement operations.

Integer Arithmetic Operations

The arithmetic operations include the four basic operations of add (ADD), subtract (SUB), multiply (MUL), and divide (DIV), as well as arithmetic compare (CMP), clear (CLR), and negate (NEG). The add and subtract instructions are available for both address and data operations, with data operations accepting all operand sizes. Address operations are limited to legal address size operands (16 or 32 bits). Data, address, and memory compare operations are also available. The clear and negate instructions may be used on all sizes of data operands.

The multiply and divide operations are available for signed and unsigned operands using word multiply to produce a long-word product, and a long-word dividend with word divisor to produce a word quotient with a word remainder.

Multiprecision and mixed size arithmetic can be accomplished using a set of extended instructions. These instructions are: add extended (ADDX), subtract extended (SUBX), sign extend (EXT), and negate binary with extend (NEGX).

A test operand (TST) instruction that will set the condition codes as a result of a compare of the operand with zero is also available. Test and set (TAS) is a synchronization instruction useful in multiprocessor systems. Table 6 is a summary of the integer arithmetic operations.

Table 6. Integer Arithmetic Operations

INSTRUCTION	OPERAND SIZE	OPERATION
ADD	8, 16, 32	Dn + (EA) → Dn
		(EA) + Dn → (EA)
		(EA) + #xxx → (EA)
	16, 32	An + (EA) → An
ADDX	8, 16, 32	Dx + Dy + X → Dx
	16, 32	–(Ax) + –(Ay) + X → (Ax)
CLR	8, 16, 32	0 → EA
CMP	8, 16, 32	Dn – (EA)
		(EA) – #xxx
		(Ax) + –(Ay)–
	16, 32	An – (EA)
DIVS	32 ÷ 16	Dn ÷ (EA) → Dn
DIVU	32 ÷ 16	Dn ÷ (EA) → Dn
EXT	8 → 16	$(Dn)_8 \rightarrow Dn_{16}$
	16 → 32	$(Dn)_{16} \rightarrow Dn_{32}$
MULS	16 × 16 → 32	Dn × (EA) → Dn
MULU	16 × 16 → 32	Dn × (EA) → Dn
NEG	8, 16, 32	0 – (EA) → (EA)
NEGX	8, 16, 32	0 – (EA) – X → (EA)
SUB	8, 16, 32	Dn – (EA) → Dn
		(EA) – Dn → (EA)
		(EA) – #xxx → (EA)
	16, 32	An – (EA) → An
SUBX	8, 16, 32	Dx – Dy – X → Dx
		–(Ax) – –(Ay) – X → (Ax)
TAS	8	(EA) – 0, 1 → EA[7]
TST	8, 16, 32	(EA) – 0

NOTES:

[ ] = bit number

= immediate data

–( ) = indirect with predecrement

( )+ = indirect with postdecrement

Logical Operations

Logical operation instructions AND, OR, EOR, and NOT are available for all sizes of integer data operands. A similar set of immediate instructions (ANDI, ORI, and EORI) provide these logical operations with all sizes of immediate data. Table 7 is a summary of the logical operations.

Table 7. Logical Operations

INSTRUCTION	OPERAND SIZE	OPERATION
AND	8, 16, 32	Dn ∧ (EA) → Dn (EA) ∧ Dn → (EA) (EA) ∧ #xxx → (EA)
OR	8, 16, 32	Dn ∨ (EA) → Dn (EA) ∨ Dn → (EA) (EA) ∨ #xxx → (EA)
EOR	8, 16, 32	(EA) ⊕ Dy → (EA) (EA) ⊕ #xxx → (EA)
NOT	8, 16, 32	~(EA) → (EA)

NOTES:

= immediate data

~ = invert

∧ = logical AND

∨ = logical OR

⊕ = logical exclusive OR

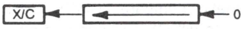
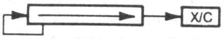
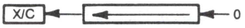
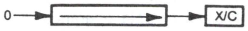
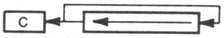
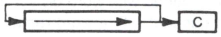
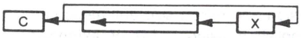
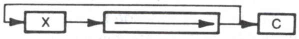
Shift and Rotate Operations

Shift operations in both directions are provided by the arithmetic instructions ASR and ASL and logical shift instructions LSR and LSL. The rotate instructions (with and without extend) available are ROXR, ROXL, ROR, and ROL. All shift and rotate operations can be performed in either registers or memory. Register shifts and rotates support all operand sizes and allow a shift count specified in a data register.

Memory shifts and rotates are for word operands only and allow only single-bit shifts or rotates.

Table 8 is a summary of the shift and rotate operations.

Table 8. Shift and Rotate Operations

INSTRUCTION	OPERAND SIZE	OPERATION
ASL	8, 16, 32	
ASR	8, 16, 32	
LSL	8, 16, 32	
LSR	8, 16, 32	
ROL	8, 16, 32	
ROR	8, 16, 32	
ROXL	8, 16, 32	
ROXR	8, 16, 32	

Bit Manipulation Operations

Bit manipulation operations are accomplished using the following instructions: bit test (BTST), bit test and set (BSET), bit test and clear (BCLR), and bit test and change (BCHG). Table 9 is a summary of the bit manipulation operations. (Z is bit 2 of the status register.)

Table 9. Bit Manipulation Operations

INSTRUCTION	OPERAND SIZE	OPERATION
BTST	8, 32	~bit of (EA) → Z
BSET	8, 32	~bit of (EA) → Z 1 → bit of EA
BCLR	8, 32	~bit of (EA) → Z 0 → bit of EA
BCHG	8, 32	~bit of (EA) → Z ~bit of (EA) → bit of EA

NOTE:
~ = invert

Binary Coded Decimal Operations

Multiprecision arithmetic operations on binary coded decimal numbers are accomplished using the following instructions: add decimal with extend (ABCD), subtract decimal with extend (SBCD), and negate decimal with extend (NBCD). Table 10 is a summary of the binary coded decimal operations.

Table 10. Binary Coded Decimal Operations

INSTRUCTION	OPERAND SIZE	OPERATION
ABCD	8	$Dx_{10} + Dy_{10} + X \rightarrow Dx$ $-(Ax)_{10} + -(Ay)_{10} + X \rightarrow (Ax)$
SBCD	8	$Dx_{10} - Dy_{10} - X \rightarrow Dx$ $-(Ax)_{10} - -(Ay)_{10} - X \rightarrow (Ax)$
NBCD	8	$0 - (EA)_{10} - X \rightarrow (EA)$

NOTE:
–( ) = indirect with predecrement.

Program Control Operations

Program control operations are accomplished using a series of conditional and unconditional branch instructions and return instructions. These instructions are summarized in Table 11.

The conditional instructions provide setting and branching for the following conditions:

CC – carry clear
CS – carry set
EQ – equal
F – never true
GE – greater or equal
GT – greater than
HI – high
LE – less or equal
LS – low or same
LT – less than
MI – minus
NE – not equal
PL – plus
T – always true
VC – no overflow
VS – overflow

Table 11. Program Control Operations

INSTRUCTION	OPERATION
Conditional	
B_{CC}	Branch conditionally (14 conditions) 8- and 16-bit displacement
DB_{CC}	Test condition, decrement, and branch 16-bit displacement
S_{CC}	Set byte conditionally (16 conditions)
Unconditional	
BRA	Branch always 8- and 16-bit displacement
BSR	Branch to subroutine 8- and 16-bit displacement
JMP	Jump
JSR	Jump to subroutine
Returns	
RTR	Return and restore condition codes
RTS	Return from subroutine

System Control Operations

System control operations are accomplished by using privileged instructions, trap generating instructions, and instructions that use or modify the status register. These instructions are summarized in Table 12.

Table 12. System Control Operations

INSTRUCTION	OPERATION
Privileged	
ANDI to SR	Logical AND to status register
EORI to SR	Logical EOR to status register
MOVE EA to SR	Load new status register
MOVE USP	Move user stack pointer
ORI to SR	Logical OR to status register
RESET	Reset external devices
RTE	Return from exception
STOP	Stop program execution
Trap generating	
CHK	Check data register against upper bounds
TRAP	Trap
TRAPV	Trap on overflow
Status register	
ANDI to CCR	Logical AND to condition codes
EORI to CCR	Logical EOR to condition codes
MOVE EA to CCR	Load new condition codes
MOVE SR to EA	Store status register
ORI to CCR	Logical OR to condition codes

SIGNAL AND BUS OPERATION DESCRIPTION

This section contains a brief description of the input and output signals. A discussion of bus operation during the various machine cycles and operations is also given.

NOTE:
The terms **assertion** and **negation** will be used extensively. This is done to avoid confusion when dealing with a mixture of "active-low" and "active-high" signals. The term assert or assertion is used to indicate that a signal is active or true, independent of whether that level is represented by a high or low voltage. The term negate or negation is used to indicate that a signal is inactive or false.

Signal Description

The input and output signals can be functionally organized into the groups shown in Figure 8. The following paragraphs provide a brief description of the signals and a reference (if applicable) to other paragraphs that contain more detail about the function being performed.

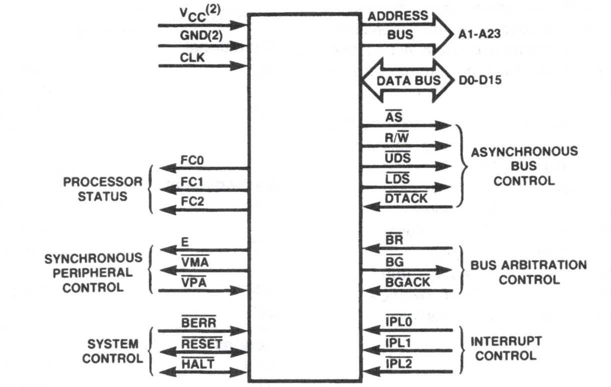


Figure 8. Input and Output Signals

Table 13. Data Strobe Control of Data Bus

$\overline{UDS}$	$\overline{LDS}$	R/$\overline{W}$	D8 – D15	D0 – D7
High	High	–	No valid data	No valid data
Low	Low	High	Valid data bits 8 – 15	Valid data bits 0 – 7
High	Low	High	No valid data	Valid data bits 0 – 7
Low	High	High	Valid data bits 8 – 15	No valid data
Low	Low	Low	Valid data bits 8 – 15	Valid data bits 0 – 7
High	Low	Low	Valid data bits 0 – 7*	Valid data bits 0 – 7
Low	High	Low	Valid data bits 8 – 15	Valid data bits 8 – 15*

NOTE:
*These conditions are a result of current implementation and may not appear on future devices.

Address Bus (A1 through A23)

This 23-bit, unidirectional, 3-State bus is capable of addressing 8 megawords of data. It provides the address for bus operation during all cycles except interrupt cycles. During interrupt cycles, address lines A1, A2, and A3 provide information about what level interrupt is being serviced while address lines A4 through A23 are all set to a logic high.

Data Bus (D0 through D15)

This 16-bit bidirectional, 3-State bus is the general purpose data path. It can transfer and accept data in either word or byte length. During an interrupt acknowledge cycle, the external device supplies the vector number on data lines D0 – D7.

Asynchronous Bus Control

Asynchronous data transfers are handled using the following control signals: address strobe, read/write, upper and lower data strobes, and data transfer acknowledge. These signals are explained in the following paragraphs.

Address Strobe ($\overline{AS}$) — This signal indicates that there is a valid address on the address bus.

Read/Write (R/$\overline{W}$) — This signal defines the data bus transfer as a read or write cycle. The R/$\overline{W}$ signal also works in conjunction with the data strobes as explained in the following paragraph.

Upper and Lower Data Strobe ($\overline{UDS}$, $\overline{LDS}$) — These signals control the flow of data on the data bus, as shown in Table 13. When the R/$\overline{W}$ line is high, the processor will read from the data bus as indicated. When the R/$\overline{W}$ line is low, the processor will write to the data bus as shown.

Data Transfer Acknowledge ($\overline{DTACK}$) — This input indicates that the data transfer is completed. When the processor recognizes $\overline{DTACK}$ during a read cycle, data is latched and the bus cycle terminated. When $\overline{DTACK}$ is recognized during a write cycle, the bus cycle is terminated. (Refer to **Asynchronous Versus Synchronous Operation**).

Bus Arbitration Control

The three signals, bus request, bus grant, and bus grant acknowledge, form a bus arbitration circuit to determine which device will be the bus master device.

Bus Request ($\overline{BR}$) — This input is wire ORed with all other devices that could be bus masters. This input indicates to the processor that some other device desires to become the bus master.

Bus Grant ($\overline{BG}$) — This output indicates to all other potential bus master devices that the processor will release bus control at the end of the current bus cycle.

Bus Grant Acknowledge ($\overline{BGACK}$) — This input indicates that some other device has become the bus master. This signal should not be asserted until the following four conditions are met:

1. a bus grant has been received,
2. address strobe is inactive which indicates that the microprocessor is not using the bus,
3. data transfer acknowledge is inactive which indicates that neither memory nor peripherals are using the bus, and
4. bus grant acknowledge is inactive which indicates that no other device is still claiming bus mastership.

Interrupt Control ($\overline{\text{IPL0}}$ $\overline{\text{IPL1}}$, $\overline{\text{IPL2}}$)
These input pins indicate the encoded priority level of the device requesting an interrupt. Level seven is the highest priority while level zero indicates that no interrupts are requested. Level seven cannot be masked. The least significant bit is given in $\overline{\text{IPL0}}$ and the most significant bit is contained in $\overline{\text{IPL2}}$. These lines must remain stable until the processor signals interrupt acknowledge (FC0 – FC2 are all high) to insure that the interrupt is recognized.

System Control
The system control inputs are used to either reset or halt the processor and to indicate to the processor that bus errors have occurred. The three system control inputs are explained in the following paragraphs.

Bus Error ($\overline{\text{BERR}}$) — This input informs the processor that there is a problem with the cycle currently being executed. Problems may be a result of:

1. nonresponding devices,
2. interrupt vector number acquisition failure,
3. illegal access request as determined by a memory management unit, or
4. other application dependent errors.

The bus error signal interacts with the halt signal to determine if the current bus cycle should be re-executed or if exception processing should be performed.

Refer to **Bus Error and Halt Operation** for additional information about the interaction of the bus error and halt signals.

Reset ($\overline{\text{RESET}}$) — This bidirectional signal line acts to reset (start a system initialization sequence) the processor in response to an external reset signal. An internally generated reset (result of a $\overline{\text{RESET}}$ instruction) causes all external devices to be reset and the internal state of the processor is not affected. A total system reset (processor and external devices) is the result of external $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ signals applied at the same time. Refer to **Reset Operation** for further information.

Halt ($\overline{\text{HALT}}$) — When this bidirectional line is driven by an external device, it will cause the processor to stop at the completion of the current bus cycle. When the processor has been halted using this input, all control signals are inactive and all 3-State lines are put in their high-impedance state (refer to Table 15). Refer to **Bus Error and Halt Operation** for additional information about the interaction between the $\overline{\text{HALT}}$ and bus error signals.

When the processor has stopped executing instructions, such as in a double bus fault condition (refer to **Double Bus Faults**), the $\overline{\text{HALT}}$ line is driven by the processor to indicate to external devices that the processor has stopped.

Peripheral Control
These control signals are used to allow the interfacing of synchronous peripheral devices with the asynchronous SCN68000. These signals are explained in the following paragraphs.

Enable (E) — This signal is the standard enable signal common to all synchronous type peripheral devices. The period for this output is ten SCN68000 clock periods (six clocks low, four clocks high). Enable is generated by an internal ring counter which may come up in any state (i.e., at power on, it is impossible to guarantee phase relationship of E to CLK). E is a free-running clock and runs regardless of the state of the bus on the MPU.

Valid Peripheral Address ($\overline{\text{VPA}}$) — This input indicates that the device or region addressed is a synchronous family device and that data transfer should be synchronized with the enable (E) signal. This input also indicates that the processor should use automatic vectoring for an interrupt. Refer to **Interface with Synchronous Peripherals.**

Valid Memory Address ($\overline{\text{VMA}}$) — This output is used to indicate to synchronous peripheral devices that there is a valid address on the address bus and the processor is synchronized to enable. This signal only responds to a valid peripheral address ($\overline{\text{VPA}}$) input which indicates that the peripheral is a synchronous family device.

Processor Status (FC0, FC1, FC2)
These function code outputs indicate the state (user or supervisor) and the cycle type currently being executed, as shown in Table 14. The information indicated by the function code outputs is valid whenever address strobe ($\overline{\text{AS}}$) is active.

Table 14. Function Code Outputs

FUNCTION CODE OUTPUT			CYCLE TYPE
FC2	FC1	FC0	
Low	Low	Low	(Undefined, reserved)
Low	Low	High	User data
Low	High	Low	User program
Low	High	High	(Undefined, reserved)
High	Low	Low	(Undefined, reserved)
High	Low	High	Supervisor data
High	High	Low	Supervisor program
High	High	High	Interrupt acknowledge

Clock (CLK)
The clock input is a TTL-compatible signal that is internally buffered for development of the internal clocks needed by the processor. The clock input should not be gated off at any time and the clock signal must conform to minimum and maximum pulse width times.

Signal Summary
Table 15 is a summary of all the signals discussed in the previous paragraphs.

Bus Operation

The following paragraphs explain control signal and bus operation during data transfer operations, bus arbitration, bus error and halt conditions, and reset operation.

Data Transfer Operations
Transfer of data between devices involves the following leads:

1. address bus A1 through A23,
2. data bus D0 through D15, and
3. control signals.

The address and data buses are separate parallel buses used to transfer data using an asynchronous bus structure. In all cycles, the bus master assumes responsibility for deskewing all signals it issues at both the start and end of a cycle. In addition, the bus master is responsible for deskewing the acknowledge and data signals from the slave device.

The following paragraphs explain the read, write, and read-modify-write cycles. The indivisible read-modify-write cycle is the method used by the SCN68000 for interlocked multiprocessor communications.

Read Cycle — During a read cycle, the processor receives data from the memory or a peripheral device. The processor reads bytes of data in all cases. If the instruction specifies a word (or double word) operation, the processor reads both upper and lower bytes simultaneously by asserting both upper and lower data strobes. When the instruction specifies byte operation, the processor uses an internal A0 bit to determine which byte to read and then issues the data strobe required for that byte. For byte operations, when the A0 bit equals zero, the upper data strobe is issued. When the A0 bit equals one, the lower data strobe is issued. When the data is received, the processor correctly positions it internally.

A word read cycle flowchart is given in figure 9. A byte read cycle flowchart is give in Figure 10. Read cycle timing is given in Figure 11. Figure 12 details word and byte read cycle operations.

Table 15. Signal Summary

SIGNAL NAME	MNEMONIC	INPUT/OUTPUT	ACTIVE STATE	Hi-Z	
				on $\overline{\text{HALT}}$	on $\overline{\text{BGACK}}$
Address bus	A1 – A23	Output	High	Yes	Yes
Data bus	D0 – D15	Input/output	High	Yes	Yes
Address strobe	$\overline{\text{AS}}$	Output	Low	No	Yes
Read/write	R/$\overline{\text{W}}$	Output	Read-high Write-low	No	Yes
Upper and lower data strobes	$\overline{\text{UDS}}$, $\overline{\text{LDS}}$	Output	Low	No	Yes
Data transfer acknowledge	$\overline{\text{DTACK}}$	Input	Low	No	No
Bus request	$\overline{\text{BR}}$	Input	Low	No	No
Bus grant	$\overline{\text{BG}}$	Output	Low	No	No
Bus grant acknowledge	$\overline{\text{BGACK}}$	Input	Low	No	No
Interrupt priority level	$\overline{\text{IPL0}}$, $\overline{\text{IPL1}}$, $\overline{\text{IPL2}}$	Input	Low	No	No
Bus error	$\overline{\text{BERR}}$	Input	Low	No	No
Reset	$\overline{\text{RESET}}$	Input/output	Low	No^1	No^1
Halt	$\overline{\text{HALT}}$	Input/output	Low	No^1	No^1
Enable	E	Output	High	No	No
Valid memory address	$\overline{\text{VMA}}$	Output	Low	No	Yes
Valid peripheral address	$\overline{\text{VPA}}$	Input	Low	No	No
Function code output	FC0, FC1, FC2	Output	High	No^2	Yes
Clock	CLK	Input	High	No	No
Power input	V_{CC}	Input	–	–	–
Ground	GND	Input	–	–	–

NOTES:

1. Open Drain
2. Function codes are placed in high-impedance state during HALT.

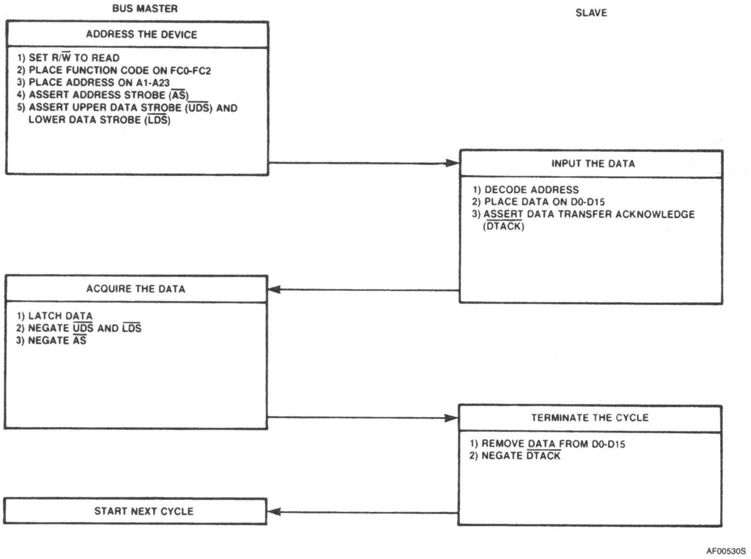

BUS MASTER
SLAVE
ADDRESS THE DEVICE
1) SET R/$\overline{W}$ TO READ
2) PLACE FUNCTION CODE ON FC0-FC2
3) PLACE ADDRESS ON A1-A23
4) ASSERT ADDRESS STROBE ($\overline{AS}$)
5) ASSERT UPPER DATA STROBE ($\overline{UDS}$) AND LOWER DATA STROBE ($\overline{LDS}$)
INPUT THE DATA
1) DECODE ADDRESS
2) PLACE DATA ON D0-D15
3) ASSERT DATA TRANSFER ACKNOWLEDGE ($\overline{DTACK}$)
ACQUIRE THE DATA
1) LATCH DATA
2) NEGATE $\overline{UDS}$ AND $\overline{LDS}$
3) NEGATE $\overline{AS}$
TERMINATE THE CYCLE
1) REMOVE DATA FROM D0-D15
2) NEGATE $\overline{DTACK}$
START NEXT CYCLE
AF00530S

Figure 9. Word Read Cycle Flowchart

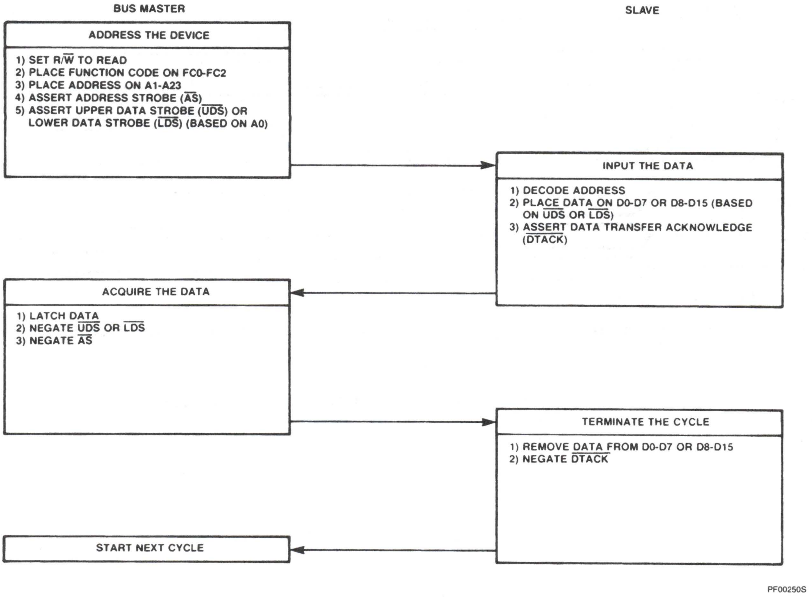

BUS MASTER
SLAVE
ADDRESS THE DEVICE
1) SET R/$\overline{W}$ TO READ
2) PLACE FUNCTION CODE ON FC0-FC2
3) PLACE ADDRESS ON A1-A23
4) ASSERT ADDRESS STROBE ($\overline{AS}$)
5) ASSERT UPPER DATA STROBE ($\overline{UDS}$) OR LOWER DATA STROBE ($\overline{LDS}$) (BASED ON A0)
INPUT THE DATA
1) DECODE ADDRESS
2) PLACE DATA ON D0-D7 OR D8-D15 (BASED ON $\overline{UDS}$ OR $\overline{LDS}$)
3) ASSERT DATA TRANSFER ACKNOWLEDGE ($\overline{DTACK}$)
ACQUIRE THE DATA
1) LATCH DATA
2) NEGATE $\overline{UDS}$ OR $\overline{LDS}$
3) NEGATE $\overline{AS}$
TERMINATE THE CYCLE
1) REMOVE DATA FROM D0-D7 OR D8-D15
2) NEGATE $\overline{DTACK}$
START NEXT CYCLE
PF00250S

Figure 10. Byte Read Cycle Flowchart

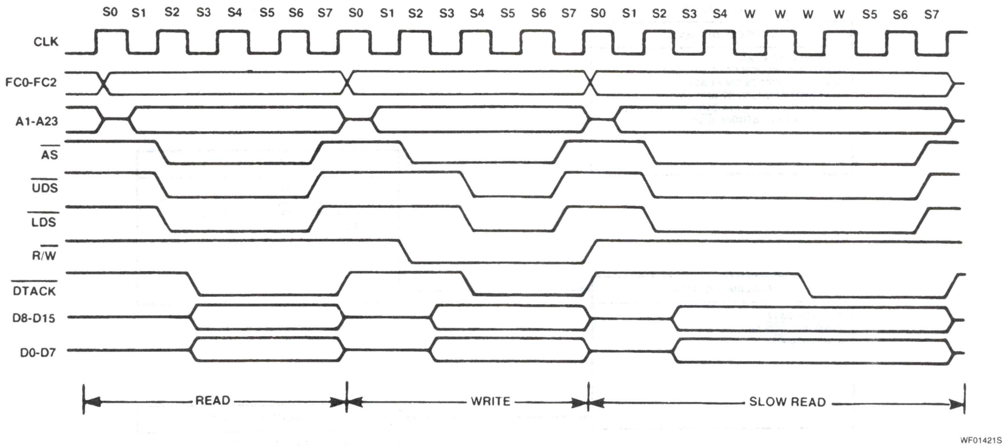

S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 W W W W S5 S6 S7
CLK
FC0-FC2
A1-A23
AS
UDS
LDS
R/W
DTACK
D8-D15
D0-D7
READ
WRITE
SLOW READ
WF01421S

Figure 11. Read and Write Cycle Timing Diagram

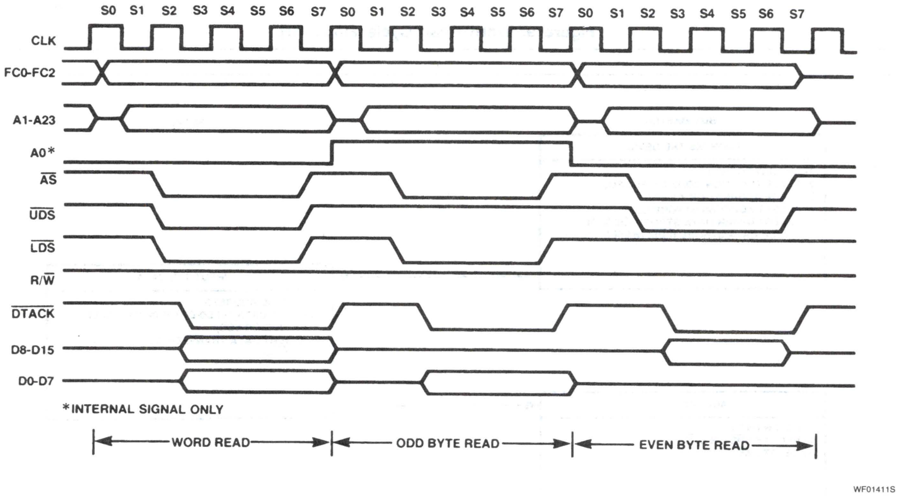

S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7
CLK
FC0-FC2
A1-A23
A0*
AS
UDS
LDS
R/W
DTACK
D8-D15
D0-D7
*INTERNAL SIGNAL ONLY
WORD READ
ODD BYTE READ
EVEN BYTE READ
WF01411S

Figure 12. Word and Byte Read Cycle Timing Diagram

Write Cycle — During a write cycle, the processor sends data to either the memory or a peripheral device. The processor writes bytes of data in all cases. If the instruction specifies a word operation, the processor writes both bytes. When the instruction specifies a byte operation, the processor uses an internal A0 bit to determine which byte to write and then issues the data strobe required for that byte. For byte operations, when the A0 bit equals zero, the upper data strobe is issued. When the A0 bit equals one, the lower data strobe is issued. A word write cycle flowchart is given in Figure 13. A byte write cycle flowchart is given in Figure 14. Write cycle timing is given in Figure 11. Figure 15 details word and byte write cycle operation.

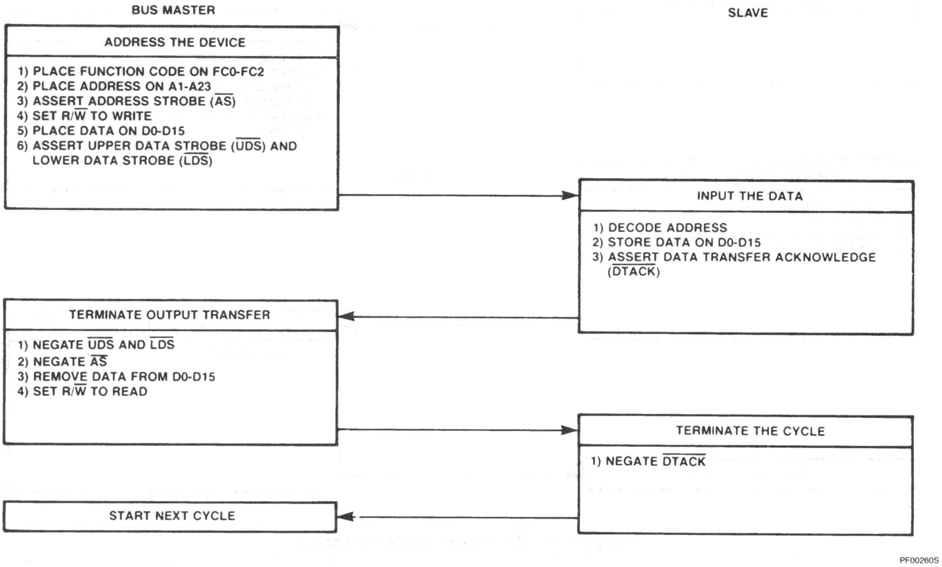

BUS MASTER
SLAVE
ADDRESS THE DEVICE
1) PLACE FUNCTION CODE ON FC0-FC2
2) PLACE ADDRESS ON A1-A23
3) ASSERT ADDRESS STROBE ($\overline{AS}$)
4) SET R/$\overline{W}$ TO WRITE
5) PLACE DATA ON D0-D15
6) ASSERT UPPER DATA STROBE ($\overline{UDS}$) AND LOWER DATA STROBE ($\overline{LDS}$)
INPUT THE DATA
1) DECODE ADDRESS
2) STORE DATA ON D0-D15
3) ASSERT DATA TRANSFER ACKNOWLEDGE ($\overline{DTACK}$)
TERMINATE OUTPUT TRANSFER
1) NEGATE $\overline{UDS}$ AND $\overline{LDS}$
2) NEGATE $\overline{AS}$
3) REMOVE DATA FROM D0-D15
4) SET R/$\overline{W}$ TO READ
TERMINATE THE CYCLE
1) NEGATE $\overline{DTACK}$
START NEXT CYCLE
PF00260S

Figure 13. Word Write Cycle Flowchart

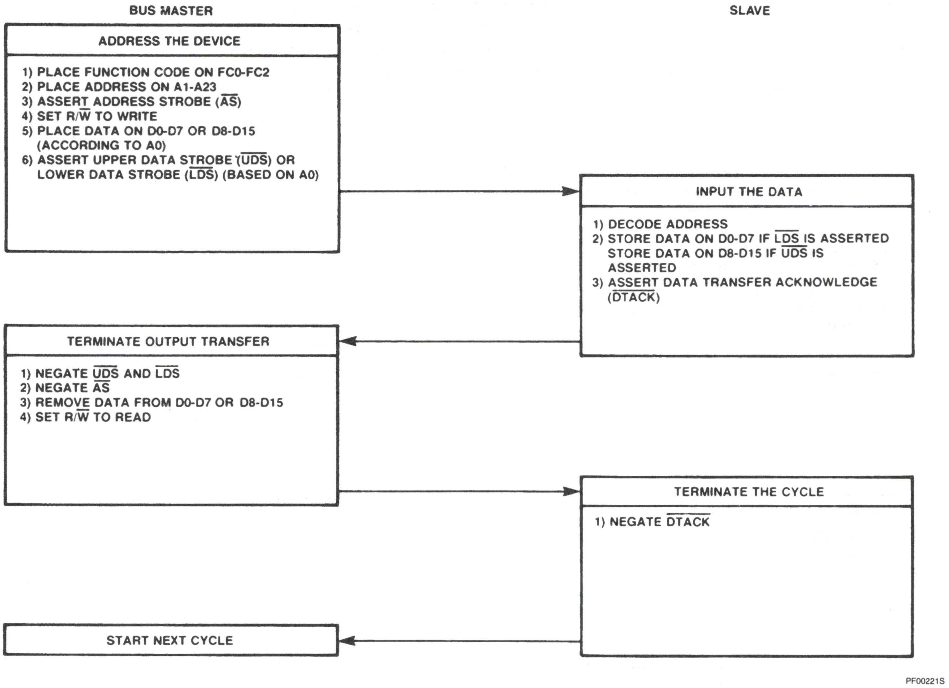

BUS MASTER
SLAVE
ADDRESS THE DEVICE
1) PLACE FUNCTION CODE ON FC0-FC2
2) PLACE ADDRESS ON A1-A23
3) ASSERT ADDRESS STROBE ($\overline{AS}$)
4) SET R/$\overline{W}$ TO WRITE
5) PLACE DATA ON D0-D7 OR D8-D15 (ACCORDING TO A0)
6) ASSERT UPPER DATA STROBE ($\overline{UDS}$) OR LOWER DATA STROBE ($\overline{LDS}$) (BASED ON A0)
INPUT THE DATA
1) DECODE ADDRESS
2) STORE DATA ON D0-D7 IF $\overline{LDS}$ IS ASSERTED STORE DATA ON D8-D15 IF $\overline{UDS}$ IS ASSERTED
3) ASSERT DATA TRANSFER ACKNOWLEDGE ($\overline{DTACK}$)
TERMINATE OUTPUT TRANSFER
1) NEGATE $\overline{UDS}$ AND $\overline{LDS}$
2) NEGATE $\overline{AS}$
3) REMOVE DATA FROM D0-D7 OR D8-D15
4) SET R/$\overline{W}$ TO READ
TERMINATE THE CYCLE
1) NEGATE $\overline{DTACK}$
START NEXT CYCLE
PF00221S

Figure 14. Byte Write Cycle Flowchart

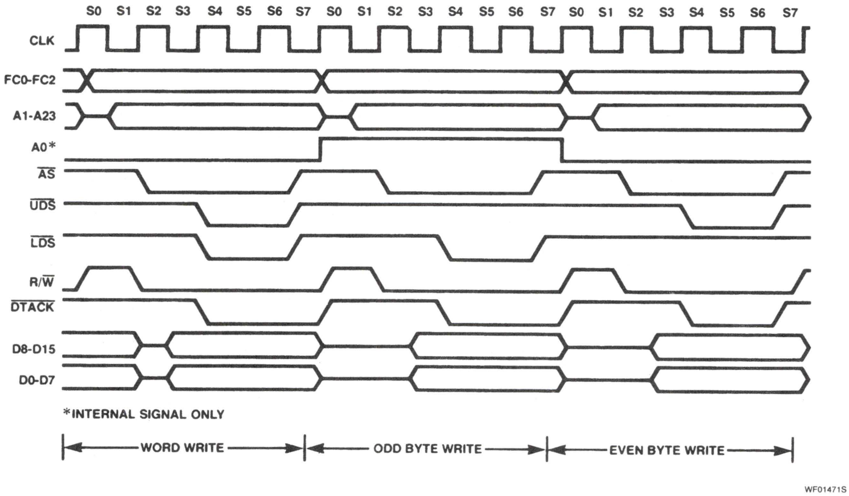

S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7
CLK
FC0-FC2
A1-A23
A0*
$\overline{AS}$
$\overline{UDS}$
$\overline{LDS}$
R/$\overline{W}$
$\overline{DTACK}$
D8-D15
D0-D7
*INTERNAL SIGNAL ONLY
WORD WRITE
ODD BYTE WRITE
EVEN BYTE WRITE
WF01471S

Figure 15. Word and Byte Write Cycle Timing Diagram

Read-Modify-Write Cycle — The read-modify-write cycle performs a read, modifies the data in the arithmetic-logic unit, and writes the data back to the same address. In the SCN68000, this cycle is indivisible in that the address strobe is asserted throughout the entire cycle. The test and set (TAS) instruction uses this cycle to provide meaningful communication between processors in a multiple processor environment. This instruction is the only instruction that uses the read-modify-write cycles and since the test and set instruction only operates on bytes, all read-modify-write cycles are byte operations. A read-modify-write cycle flowchart is given in Figure 16 and a timing diagram is given in Figure 17.

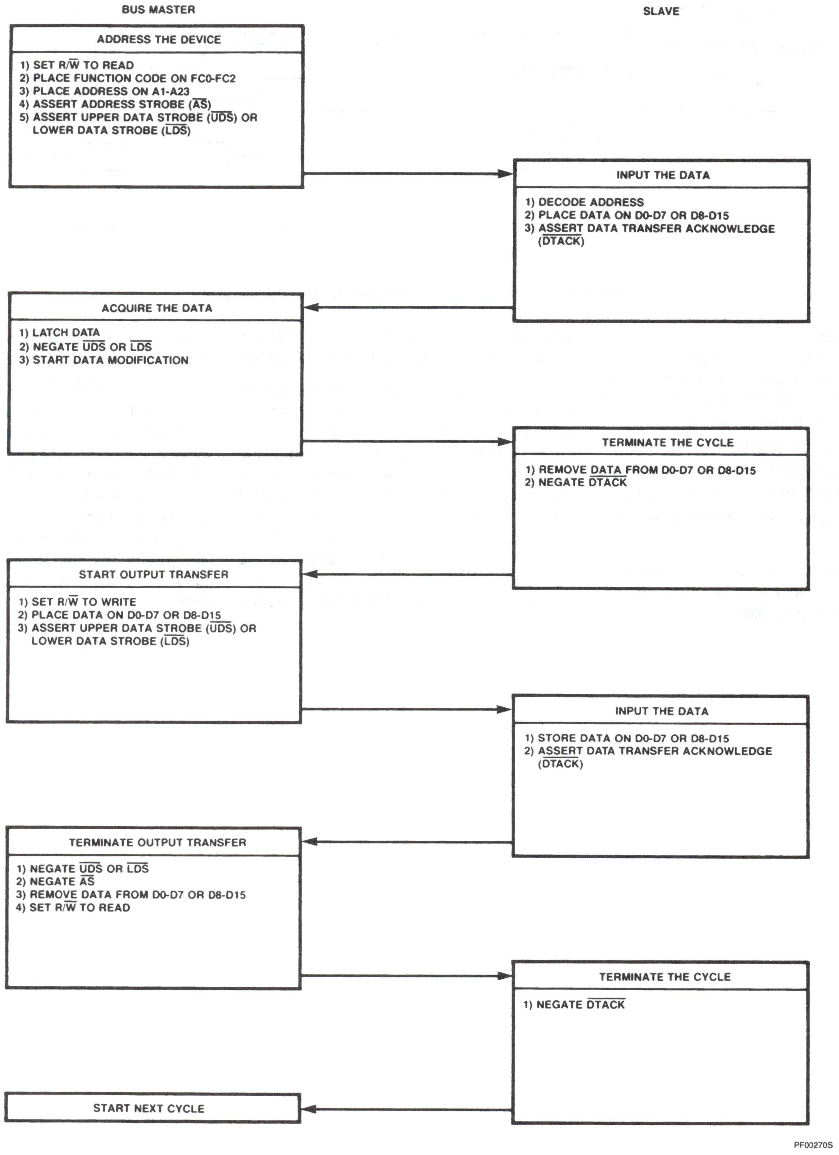

BUS MASTER
SLAVE
ADDRESS THE DEVICE
1) SET R/W̄ TO READ
2) PLACE FUNCTION CODE ON FC0-FC2
3) PLACE ADDRESS ON A1-A23
4) ASSERT ADDRESS STROBE (AS̄)
5) ASSERT UPPER DATA STROBE (UDS̄) OR LOWER DATA STROBE (LDS̄)
INPUT THE DATA
1) DECODE ADDRESS
2) PLACE DATA ON D0-D7 OR D8-D15
3) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK̄)
ACQUIRE THE DATA
1) LATCH DATA
2) NEGATE UDS̄ OR LDS̄
3) START DATA MODIFICATION
TERMINATE THE CYCLE
1) REMOVE DATA FROM D0-D7 OR D8-D15
2) NEGATE DTACK̄
START OUTPUT TRANSFER
1) SET R/W̄ TO WRITE
2) PLACE DATA ON D0-D7 OR D8-D15
3) ASSERT UPPER DATA STROBE (UDS̄) OR LOWER DATA STROBE (LDS̄)
INPUT THE DATA
1) STORE DATA ON D0-D7 OR D8-D15
2) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK̄)
TERMINATE OUTPUT TRANSFER
1) NEGATE UDS̄ OR LDS̄
2) NEGATE AS̄
3) REMOVE DATA FROM D0-D7 OR D8-D15
4) SET R/W̄ TO READ
TERMINATE THE CYCLE
1) NEGATE DTACK̄
START NEXT CYCLE
PF00270S

Figure 16. Read-Modify-Write Cycle Flowchart

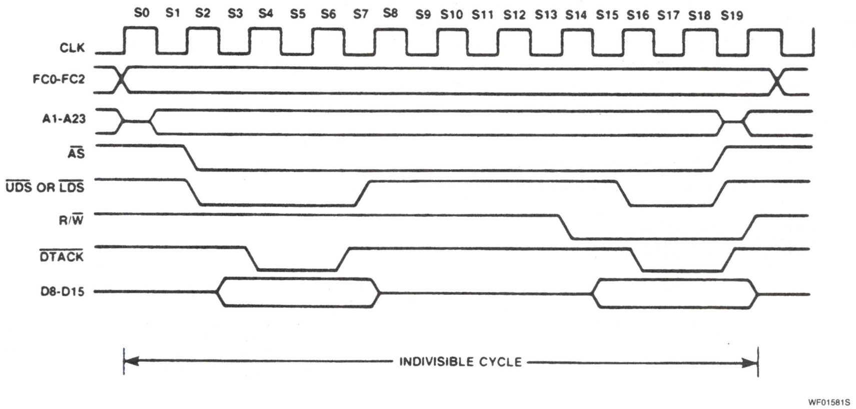

S0 S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14 S15 S16 S17 S18 S19
CLK
FC0-FC2
A1-A23
AS
UDS OR LDS
R/W
DTACK
D8-D15
INDIVISIBLE CYCLE
WF01581S

Figure 17. Read-Modify-Write Cycle Timing Diagram

Bus Arbitration

Bus arbitration is a technique used by master-type devices to request, be granted, and acknowledge bus mastership. In its simplest form, it consists of the following:

1. asserting a bus mastership request,
2. receiving a grant that the bus is available at the end of the current cycle, and
3. acknowledging that mastership has been assumed.

Figure 18 is a flowchart showing the detail involved in a request from a single device. Figure 19 is a timing diagram for the same operation. This technique allows processing of bus requests during data transfer cycles.

The timing diagram shows that the bus request is negated at the time that an acknowledge is asserted. This type of operation would be true for a system consisting of the processor and one device capable of bus mastership. In systems having a number of devices capable of bus mastership, the bus request line from each device is wire-ORed to the processor. In this system, it is easy to see that there could be more than one bus request being made. The timing diagram shows that the bus grant signal is negated a few clock cycles after the transition of the acknowledge ($\overline{BGACK}$) signal.

However, if the bus requests are still pending, the processor will assert another bus grant within a few clock cycles after it was negated. This additional assertion of bus grant allows external arbitration circuitry to select the next bus master before the current bus master has completed its requirements. The following paragraphs provide additional information about the three steps in the arbitration process.

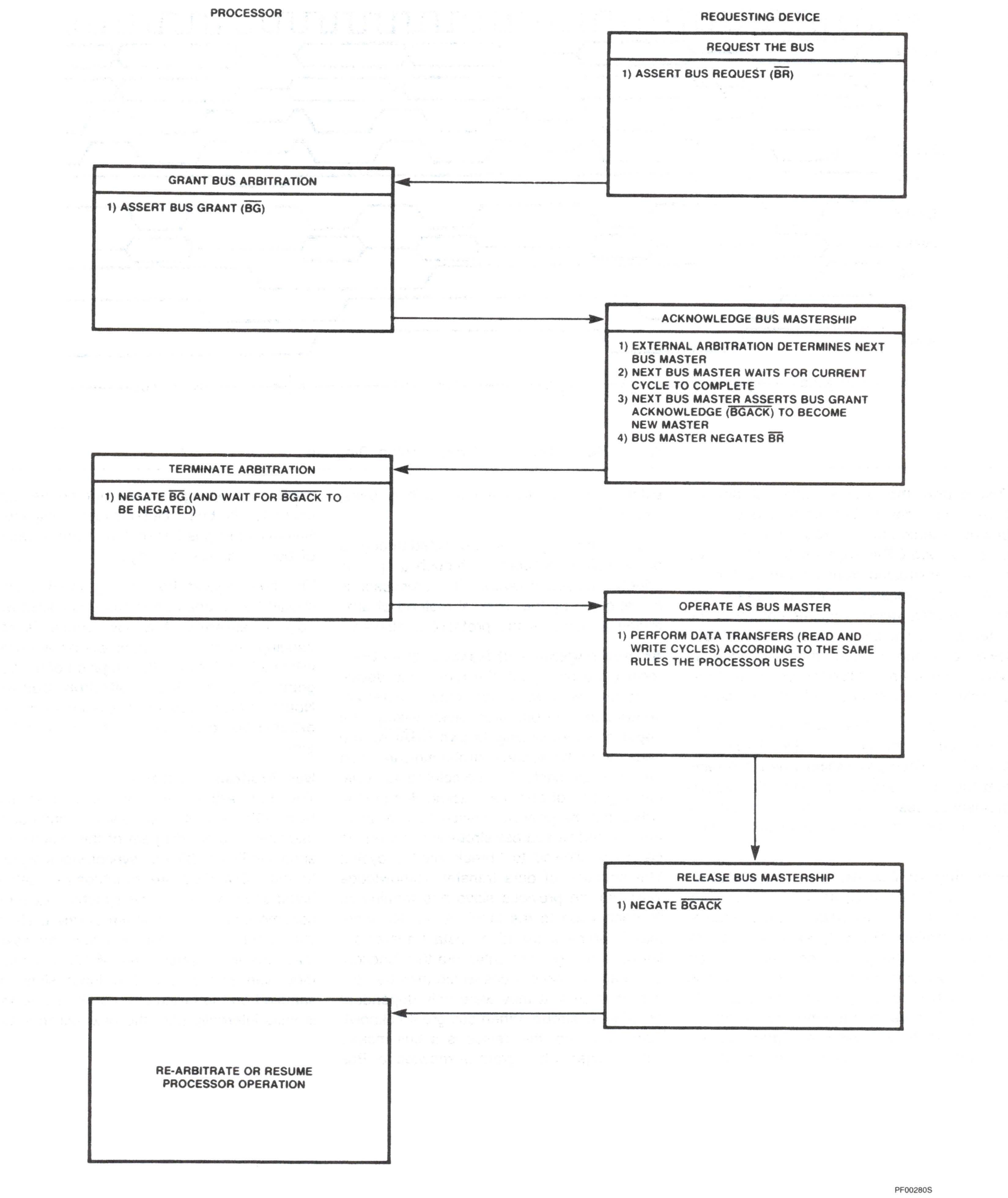

PROCESSOR
REQUESTING DEVICE
REQUEST THE BUS
1) ASSERT BUS REQUEST ($\overline{BR}$)
GRANT BUS ARBITRATION
1) ASSERT BUS GRANT ($\overline{BG}$)
ACKNOWLEDGE BUS MASTERSHIP
1) EXTERNAL ARBITRATION DETERMINES NEXT BUS MASTER
2) NEXT BUS MASTER WAITS FOR CURRENT CYCLE TO COMPLETE
3) NEXT BUS MASTER ASSERTS BUS GRANT ACKNOWLEDGE ($\overline{BGACK}$) TO BECOME NEW MASTER
4) BUS MASTER NEGATES $\overline{BR}$
TERMINATE ARBITRATION
1) NEGATE $\overline{BG}$ (AND WAIT FOR $\overline{BGACK}$ TO BE NEGATED)
OPERATE AS BUS MASTER
1) PERFORM DATA TRANSFERS (READ AND WRITE CYCLES) ACCORDING TO THE SAME RULES THE PROCESSOR USES
RELEASE BUS MASTERSHIP
1) NEGATE $\overline{BGACK}$
RE-ARBITRATE OR RESUME PROCESSOR OPERATION
PF00280S

Figure 18. Bus Arbitration Cycle Flowchart

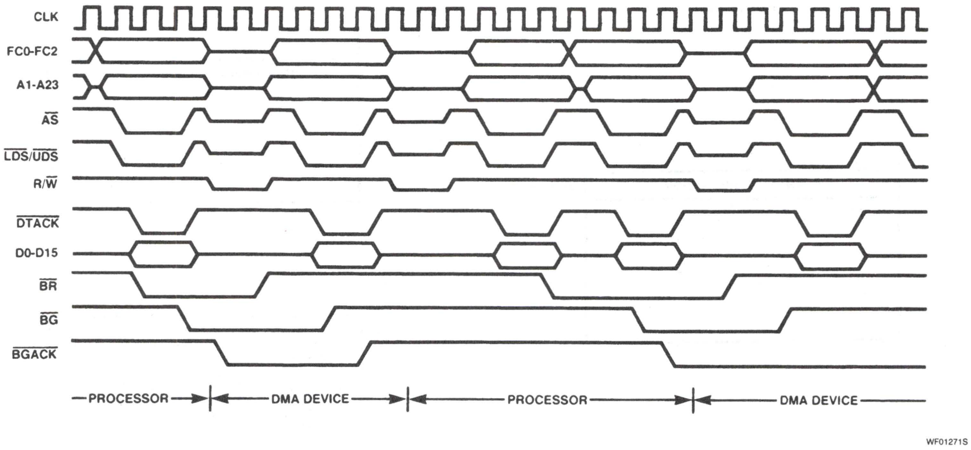


Figure 19. Bus Arbitration Cycle Timing Diagram

Requesting the Bus — External devices capable of becoming bus masters request the bus by asserting the bus request ($\overline{BR}$) signal. This is a wire-ORed signal (although it need not be constructed from open-collector devices) that indicates to the processor that some external device requires control of the external bus. The processor is effectively at a lower bus priority level than the external device and will relinquish the bus after it has completed the last bus cycle it has started.

When no acknowledge is received before the bus request signal goes inactive, the processor will continue processing when it detects that the bus request is inactive. This allows ordinary processing to continue if the arbitration circuitry responded to noise inadvertently.

Receiving the Bus Grant — The processor asserts bus grant ($\overline{BG}$) as soon as possible. Normally this is immediately after internal synchronization. The only exception to this occurs when the processor has made an internal decision to execute the next bus cycle but has not progressed far enough into the cycle to have asserted the address strobe ($\overline{AS}$) signal. In this case, bus grant will be delayed until $\overline{AS}$ is asserted to indicate to external devices that a bus cycle is being executed.

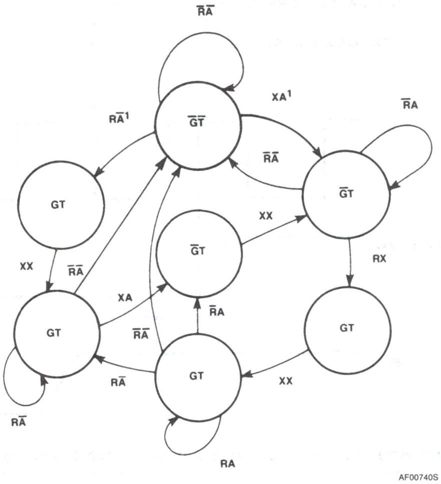
The bus grant signal may be routed through a daisy-chained network or through a specific priority-encoded network. The processor is not affected by the external method of arbitration as long as the protocol is obeyed.

Acknowledgement of Mastership — Upon receiving a bus grant, the requesting device waits until address strobe, data transfer acknowledge, and bus grant acknowledge are negated before issuing its own $\overline{BGACK}$. The negation of the address strobe indicates that the previous master has completed its cycle; the negation of bus grant acknowledge indicates that the previous master has released the bus. (While address strobe is asserted, no device is allowed to "break into" a cycle.) The negation of data transfer acknowledge indicates the previous slave has terminated its connection to the previous master. Note that in some applications data transfer acknowledge might not enter into this function. General purpose devices would then be connected such that they were only dependent on address strobe. When bus grant acknowledge is issued, the device is a bus master until it negates bus grant acknowledge. Bus grant acknowledge should not be negated until after the bus cycle(s) is (are) completed. Bus mastership is terminated at the negation of bus grant acknowledge.

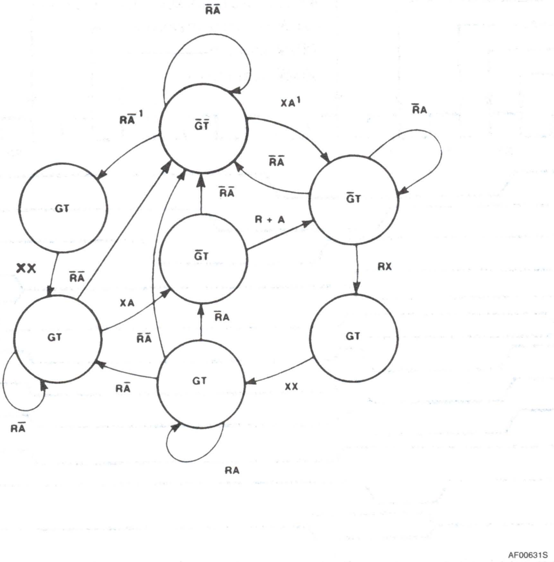
The bus request from the granted device should be dropped after bus grant acknowledge is asserted. If a bus request is still pending, another bus grant will be asserted within a few clocks of the negation of the bus grant. Refer to **Bus Arbitration Control.** Note that the processor does not perform any external bus cycles before it re-asserts bus grant.

Bus Arbitration Control

The bus arbitration control unit in the SCN68000 is implemented with a finite state machine. A state diagram of this machine is shown in Figure 20. All asynchronous signals to the SCN68000 are synchronized before being used internally. This synchronization is accomplished in a maximum of one cycle of the system clock, assuming that the asynchronous input setup time (#47) has been met (see Figure 21). The input signal is sampled on the falling edge of the clock and is valid internally after the next falling edge.


$\overline{R}\overline{A}$
XA^1
$\overline{R}A$
$R\overline{A}^1$
$\overline{G}\overline{T}$
$\overline{R}\overline{A}$
$\overline{G}T$
GT
XX
$\overline{G}T$
RX
XX
$\overline{R}\overline{A}$
XA
$\overline{R}A$
$\overline{R}\overline{A}$
GT
GT
$R\overline{A}$
GT
XX
$R\overline{A}$
RA
AF00740S

a. State Diagram for Mask Sets Previous to GN7


$\overline{R}\overline{A}$
XA^1
$\overline{R}A$
$R\overline{A}^1$
$\overline{G}\overline{T}$
$\overline{R}\overline{A}$
$\overline{G}T$
$\overline{R}\overline{A}$
GT
R + A
$\overline{G}T$
XX
RX
$\overline{R}\overline{A}$
XA
$\overline{R}A$
$\overline{R}\overline{A}$
GT
GT
GT
$R\overline{A}$
XX
$R\overline{A}$
RA
AF00631S

b. State Diagram for GN7 and Later Mask Sets

NOTES:

R = Bus request internal
A = Bus grant acknowledge internal
G = Bus grant
T = 3-State control to bus control logic[2]
X = Don't care

1. State machine will not change if the bus is S0 or S1. Refer to Bus Arbitration Control.
2. The address bus will be placed in the high-impedance state if T is asserted and $\overline{AS}$ is negated.

Figure 20. SCN68000 Bus Arbitration Unit State Diagram

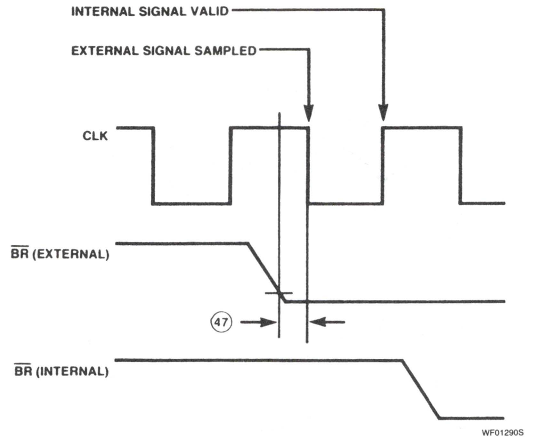

INTERNAL SIGNAL VALID
EXTERNAL SIGNAL SAMPLED
CLK
BR (EXTERNAL)
47
BR (INTERNAL)
WF01290S

Figure 21. Timing Relationship of External Asynchronous Inputs to Internal Signals

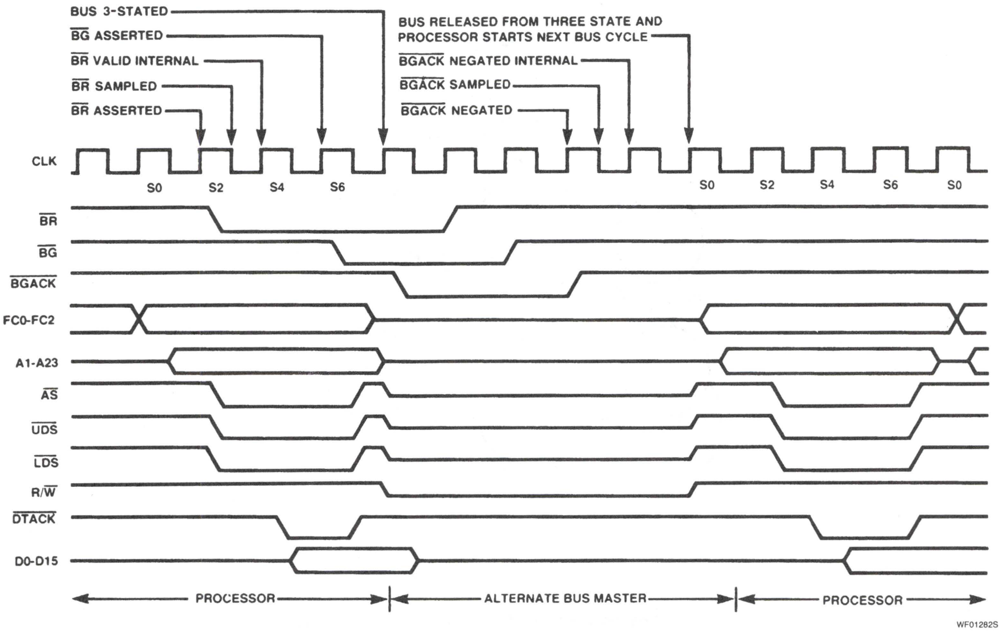

BUS 3-STATED
BG ASSERTED
BR VALID INTERNAL
BR SAMPLED
BR ASSERTED
BUS RELEASED FROM THREE STATE AND PROCESSOR STARTS NEXT BUS CYCLE
BGACK NEGATED INTERNAL
BGACK SAMPLED
BGACK NEGATED
CLK
S0
S2
S4
S6
S0
S2
S4
S6
S0
BR
BG
BGACK
FC0-FC2
A1-A23
AS
UDS
LDS
R/W
DTACK
D0-D15
PROCESSOR
ALTERNATE BUS MASTER
PROCESSOR
WF01282S

Figure 22. Bus Arbitration Timing Diagram—Processor Active

As shown in Figure 20, input signals labeled R and A are internally synchronized on the bus request and bus grant acknowledge pins, respectively. The bus grant output is labeled G and the internal 3-State control signal T. If T is true, the address, data, and control buses are placed in a high-impedance state when $\overline{AS}$ is negated. All signals are shown in positive logic (active high) regardless of their true active voltage level. State changes (valid outputs) occur on the next rising edge after the internal signal is valid.

A timing diagram of the bus arbitration sequence during a processor bus cycle is shown in Figure 22. The bus arbitration sequence while the bus is inactive (i.e., executing internal operations such as a multiply instruction) is shown in Figure 23.

If a bus request is made at a time when the MPU has already begun a bus cycle but $\overline{AS}$ has not been asserted (bus state S0), $\overline{BG}$ will not be asserted on the next rising edge. Instead, $\overline{BG}$ will be delayed until the second rising edge following its internal assertion. This sequence is shown in Figure 24.

Bus Error and Halt Operation

In a bus architecture that requires a handshake from an external device, the possibility exists that the handshake might not occur. Since different systems will require a different maximum response time, a bus error input is provided. External circuitry must be used to determine the duration between address strobe and data transfer acknowledge before issuing a bus error signal. When a bus error signal is received, the processor has two options: initiate a bus error exception sequence or try running the bus cycle again.

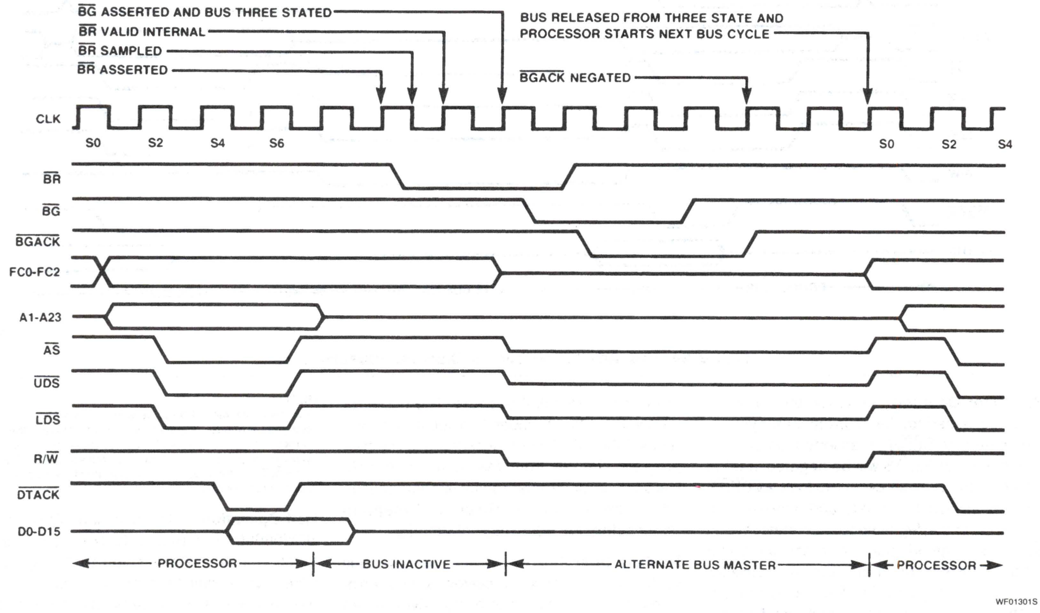

$\overline{BG}$ ASSERTED AND BUS THREE STATED
$\overline{BR}$ VALID INTERNAL
$\overline{BR}$ SAMPLED
$\overline{BR}$ ASSERTED
BUS RELEASED FROM THREE STATE AND PROCESSOR STARTS NEXT BUS CYCLE
$\overline{BGACK}$ NEGATED
CLK
S0
S2
S4
S6
S0
S2
S4
$\overline{BR}$
$\overline{BG}$
$\overline{BGACK}$
FC0-FC2
A1-A23
$\overline{AS}$
$\overline{UDS}$
$\overline{LDS}$
R/$\overline{W}$
$\overline{DTACK}$
D0-D15
PROCESSOR
BUS INACTIVE
ALTERNATE BUS MASTER
PROCESSOR
WF01301S

Figure 23. Bus Arbitration Timing Diagram—Bus Inactive

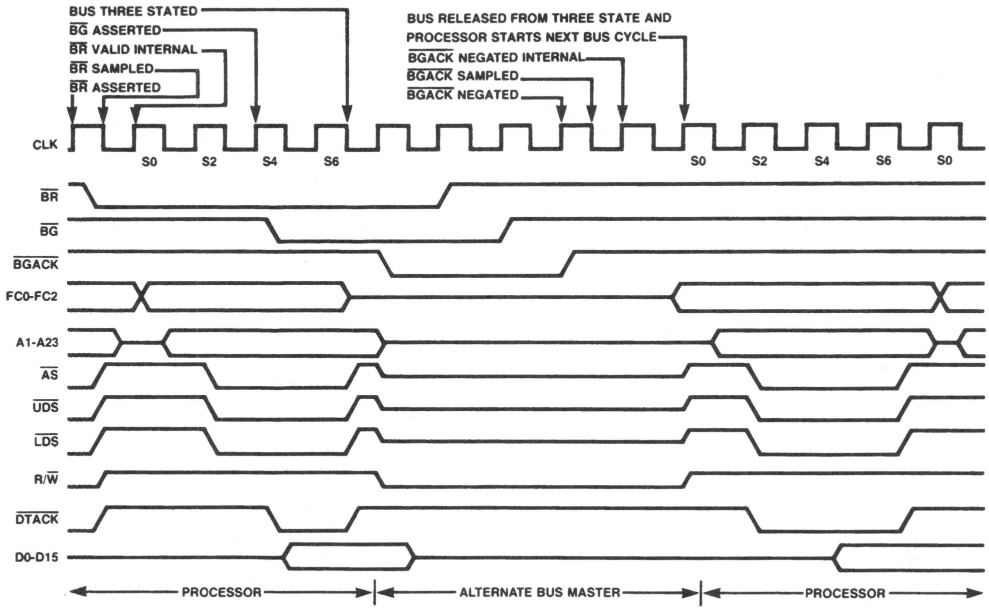


Figure 24. Bus Arbitration Timing Diagram—Special Case

Bus Error Operation — When the bus error signal is asserted, the current bus cycle is terminated. If BERR is asserted before the falling edge of S2, AS will be negated in S7 in either a read or write cycle. As long as BERR remains asserted, the data and address buses will be in the high-impedance state. When BERR is negated, the processor will begin stacking for exception processing. Figure 25 is a timing diagram for the exception sequence. The sequence is composed of the following elements:

1. stacking the program counter and status register,
2. stacking the error information,
3. reading the bus error vector table entry, and
4. executing the bus error handler routine.

The stacking of the program counter and the status register is the same as if an interrupt had occurred. Several additional items are stacked when a bus error occurs. These items are used to determine the nature of the error and correct it, if possible. The bus error vector is vector number two located at address $000008. The processor loads the new program counter from this location. A software bus error handler routine is then executed by the processor. Refer to **Exception Processing** for additional information.

Re-Run Operation — When, during a bus cycle, the processor receives a bus error signal and the halt pin is being driven by an external device, the processor enters the re-run sequence. Figure 26 is a timing diagram for re-running the bus cycle.

The processor terminates the bus cycle, then puts the address and data output lines in the high-impedance state. The processor remains "halted", and will not run another bus cycle until the halt signal is removed by external logic. Then the processor will re-run the previous cycle using the same function codes, the same data (for a write operation), and the same controls. The bus error signal should be removed at least one clock cycle before the halt signal is removed.

NOTE:
The processor will not re-run a read-modify-write cycle. This restriction is made to guarantee that the entire cycle runs correctly and that the write operation of a test-and-set operation is performed without ever releasing AS. If BERR and HALT are asserted during a read-modify-write bus cycle, a bus error operation results.

Halt Operation — The halt input signal to the SCN68000 performs a halt/run/single-step function in a similar fashion to the synchronous device halt function. The halt and run modes are somewhat self-explanatory in that when the halt signal is constantly active the processor "halts" (does nothing) and when the halt signal is constantly inactive the processor "runs" (does something).

This single-step mode is derived from correctly timed transitions on the halt signal input. It forces the processor to execute a single bus cycle by entering the run mode until the processor starts a bus cycle then changing to the halt mode. Thus, the single-step mode allows the user to proceed through (and therefore debug) processor operations one bus cycle at a time.

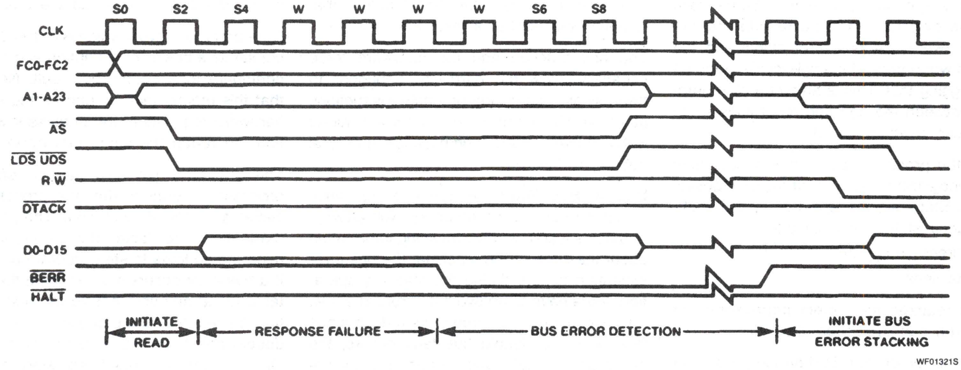

S0
S2
S4
W
W
W
W
S6
S8
CLK
FC0-FC2
A1-A23
AS
LDS UDS
R W
DTACK
D0-D15
BERR
HALT
INITIATE READ
RESPONSE FAILURE
BUS ERROR DETECTION
INITIATE BUS ERROR STACKING
WF01321S

Figure 25. Bus Error Timing Diagram

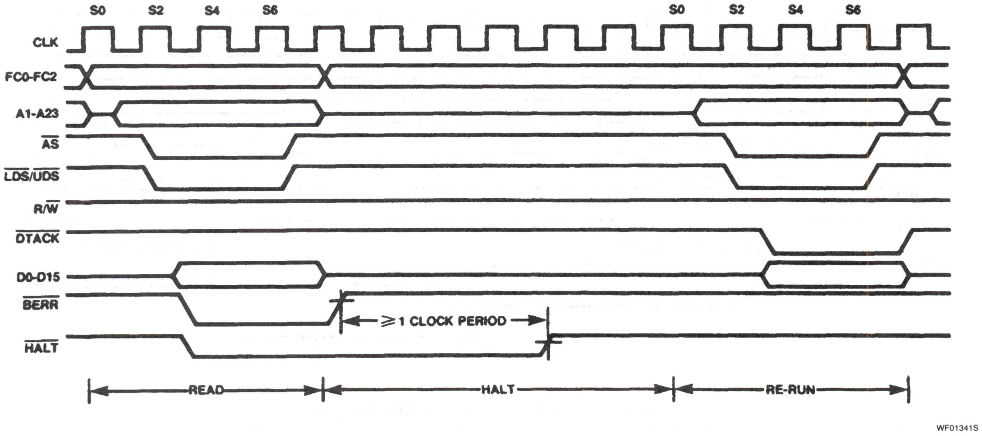

S0
S2
S4
S6
S0
S2
S4
S6
CLK
FC0-FC2
A1-A23
AS
LDS/UDS
R/W
DTACK
D0-D15
BERR
HALT
≥1 CLOCK PERIOD
READ
HALT
RE-RUN
WF01341S

Figure 26. Re-Run Bus Cycle Timing Diagram

Figure 27 details the timing required for correct single-step operations. Some care must be exercised to avoid harmful interactions between the bus error signal and the halt pin when using the single-cycle mode as a debugging tool. This is also true of interactions between the halt and reset lines since these can reset the machine.

When the processor completes a bus cycle after recognizing that the halt signal is active, most 3-State signals are put in the high-impedance state, these include:

1. address lines, and
2. data lines.

This is required for correct performance of the re-run bus cycle operation.

While the processor is honoring the halt request, bus arbitration performs as usual. That is, halting has no effect on bus arbitration. It is the bus arbitration function that removes the control signals from the bus.

The halt function and the hardware trace capability allow the hardware debugger to trace single bus cycles or single instructions at a time. These processor capabilities, along with a software debugging package, give total debugging flexibility.

Double Bus Faults — When a bus error exception occurs, the processor will attempt to stack several words containing information about the state of the machine. If a bus error exception occurs during the stacking operation, there have been two bus errors in a row. This is commonly referred to as a double bus fault. When a double bus fault occurs, the processor will halt. Once a bus error exception has occurred, any bus error exception occurring before the execution of the next instruction constitutes a double bus fault.

Note that a bus cycle which is re-run does not constitute a bus error exception and does not contribute to a double bus fault. Note also that this means that as long as the external hardware requests it, the processor will continue to re-run the same bus cycle.

The bus error pin also has an effect on processor operation after the processor receives an external reset input. The processor reads the vector table after a reset to determine the address to start program execution. If a bus error occurs while reading the vector table (or at any time before the first instruction is executed), the processor reacts as if a double bus fault has occurred and it halts. Only an external reset will start a halted processor.

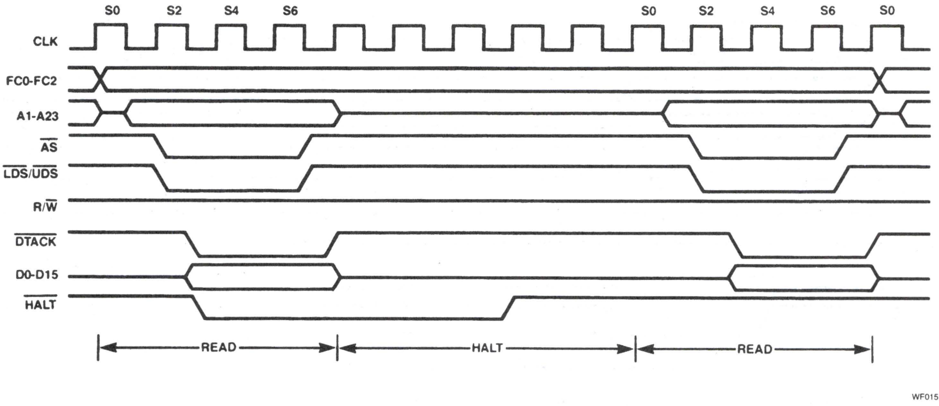


Figure 27. Halt Processor Timing Diagram

Reset Operation

The reset signal is a bidirectional signal that allows either the processor or an external signal to reset the system. Figure 28 is a timing diagram for the reset operation. Both the halt and reset lines must be asserted to ensure total reset of the processor.

When the reset and halt lines are driven by an external device, it is recognized as an entire system reset, including the processor. The processor responds by reading the reset vector table entry (vector number zero, address \$000000) and loads it into the supervisor stack pointer (SSP). Vector table entry number one at address \$000004 is read next and loaded into the program counter. The processor initializes the status register to an interrupt level of seven. No other registers are affected by the reset sequence.

When a reset instruction is executed, the processor drives the reset pin for 124 clock periods. In this case, the processor is trying to reset the rest of the system. Therefore, there is no effect on the internal state of the processor. All of the processor's internal registers and the status register are unaffected by the execution of a reset instruction. All external devices connected to the reset line will be reset at the completion of the reset instruction.

Asserting the reset and halt lines for ten clock cycles will cause a processor reset, except when V_{CC} is initially applied to the processor. In this case, an external reset must be applied for at least 100ms.

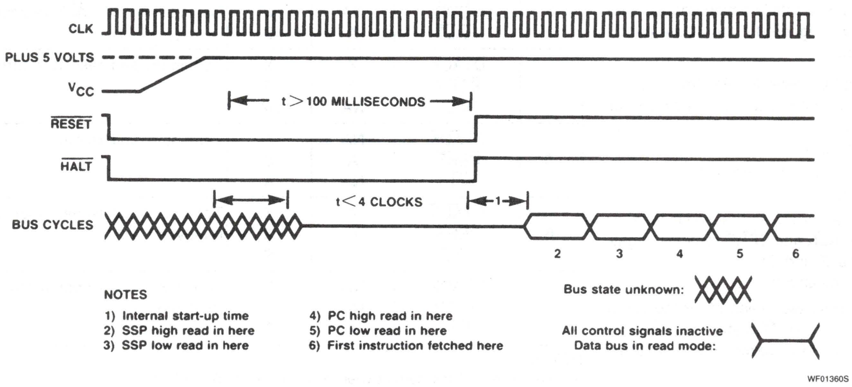

CLK
PLUS 5 VOLTS
V_{CC}
t > 100 MILLISECONDS
RESET
HALT
t < 4 CLOCKS
1
BUS CYCLES
2
3
4
5
6
NOTES
1) Internal start-up time
2) SSP high read in here
3) SSP low read in here
4) PC high read in here
5) PC low read in here
6) First instruction fetched here
Bus state unknown:
All control signals inactive
Data bus in read mode:
WF01360S

Figure 28. Reset Operation Timing Diagram

The Relationship of DTACK, BERR, and HALT

In order to properly control termination of a bus cycle for a re-run or a bus error condition, DTACK, BERR, and HALT should be asserted and negated on the rising edge of the SCN68000 clock. This will assure that when two signals are asserted simultaneously, the required setup time (#47) for both of them will be met during the same bus state.

This, or some equivalent precaution, should be designed external to the SCN68000. Parameter #48 (see AC Electrical Characteristics for # references) is intended to ensure this operation in a totally asynchronous system, and may be ignored if the above conditions are met.

The preferred bus cycle terminations may be summarized as follows (case numbers refer to Table 16):

Normal Termination:

DTACK occurs first (case 1).

Halt Termination:

HALT is asserted at the same time or before DTACK and BERR remains negated (cases 2 and 3).

Bus Error Termination:

BERR is asserted in lieu of, at the same time, or before DTACK (case 4); BERR is negated at the same time or after DTACK.

Re-Run Termination:

HALT and BERR are asserted in lieu of, at the same time, or before DTACK (cases 6 and 7); HALT must be held at least one cycle after BERR. Case 5 indicates BERR may precede HALT.

Table 16 details the resulting bus cycle termination under various combinations of control signal sequences. The negation of these same control signals under several conditions is shown in Table 17 (DTACK is assumed to be negated normally in all cases; for best results, both DTACK and BERR should be negated when address strobe is negated).

EXAMPLE A:
A system uses a watch-dog timer to terminate accesses to unpopulated address space. The timer asserts DTACK and BERR simultaneously after time out (case 4).

EXAMPLE B:
A system uses error detection on RAM contents. Designer may (a) delay DTACK until data verified and return BERR and HALT simultaneously to re-run error cycle (case 6), or if valid, return DTACK (case 1); (b) delay DTACK until data verified and return BERR at same time as DTACK if data in error (case 4).

Table 16. DTACK, BERR, and HALT Assertion Results

CASE NO.	CONTROL SIGNAL	ASSERTED ON RISING EDGE OF STATE		RESULT
		N	N + 2	
1	DTACK BERR HALT	A NA NA	S X X	Normal cycle terminate and continue.
2	DTACK BERR HALT	A NA A	S X S	Normal cycle terminate and halt. Continue when HALT removed.
3	DTACK BERR HALT	NA NA A	A NA S	Normal cycle terminate and halt. Continue when HALT removed.
4	DTACK BERR HALT	X A NA	X S NA	Terminate and take bus error trap.
5	DTACK BERR HALT	NA A NA	X S A	Terminate and re-run.
6	DTACK BERR HALT	X A A	X S S	Terminate and re-run when HALT removed.
7	DTACK BERR HALT	NA NA A	X A S	Terminate and re-run when HALT removed.

NOTES:
N – the number of the current even bus state (e.g., S4, S6, etc.)
A – signal is asserted in this bus state
NA – signal is not asserted in this state
X – don't care
S – signal was asserted in previous state and remains asserted in this state

Table 17. BERR and HALT Negation Results

CONDITIONS OF TERMINATION IN TABLE 16	CONTROL SIGNAL	NEGATED ON RISING EDGE OF STATE		RESULTS–NEXT CYCLE
		N	N+2	
Bus Error	BERR HALT	• or • or	• •	Takes bus error trap.
Re-run	BERR HALT	• or •	•	Illegal sequence; usually traps to vector number 0.
Re-run	BERR HALT	•	 •	Re-runs the bus cycle.
Normal	BERR HALT	• • or	 •	May lengthen next cycle.
Normal	BERR HALT	 • or	• none	If next cycle is started it will be terminated as a bus error.

NOTE:
• = Signal is negated in this bus state.

Asynchronous Versus Synchronous Operation

Asynchronous Operation

To achieve clock frequency independence at a system level, the SCN68000 can be used in an asynchronous manner. This entails using only the bus handshake lines (AS, UDS, LDS, DTACK, BERR, HALT, and VPA) to control the data transfer. Using this method, AS signals the start of a bus cycle and the data strobes are used as a condition for valid data on a write cycle. The slave device (memory or peripheral) then responds by placing the requested data on the data bus for a read cycle or latching data on a write cycle and asserting the data transfer acknowledge signal ($\overline{\text{DTACK}}$) to terminate the bus cycle. If no slave responds or the access is invalid, external control logic asserts the $\overline{\text{BERR}}$, or $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$, signal to abort or rerun the bus cycle.

The $\overline{\text{DTACK}}$ signal is allowed to be asserted before the data from a slave device is valid on a read cycle. The length of time that $\overline{\text{DTACK}}$ may precede data is given as parameter #31 and it must be met in any asynchronous system to insure that valid data is latched into the processor. Notice that there is no maximum time specified from the assertion of $\overline{\text{AS}}$ to the assertion of $\overline{\text{DTACK}}$. This is because the MPU will insert wait cycles of one clock period each until $\overline{\text{DTACK}}$ is recognized.

Synchronous Operation

To allow for those systems which use the system clock as a signal to generate $\overline{\text{DTACK}}$ and other asynchronous inputs, the asynchronous input setup time is given as parameter #47. If this setup is met on an input, such as $\overline{\text{DTACK}}$, the processor is guaranteed to recognize that signal on the next falling edge of the system clock. However, the converse is not true — if the input signal does not meet the setup time it is not guaranteed not to be recognized. In addition, if $\overline{\text{DTACK}}$ is recognized on a falling edge, valid data will be latched into the processor (on a read cycle) on the next falling edge provided that the data meets the setup time given as parameter #27. Given this, parameter #31 may be ignored. Note that if $\overline{\text{DTACK}}$ is asserted, with the required setup time, before the falling edge of S4, no wait states will be incurred and the bus cycle will run at its maximum speed of four clock periods.

NOTE:
During an active bus cycle, $\overline{\text{VPA}}$ and $\overline{\text{BERR}}$ is sampled on every falling edge of the clock starting with S2. $\overline{\text{DTACK}}$ is sampled on every falling edge of the clock starting with S4 and data is latched on the falling edge of S6 during a read. The bus cycle will then be terminated in S7 except when $\overline{\text{BERR}}$ is asserted in the absence of $\overline{\text{DTACK}}$, in which case it will terminate one clock cycle later in S9. $\overline{\text{VPA}}$ is sampled only on the third falling edge of the system clock before the rising edge of the E clock.

PROCESSING STATES

This section describes the actions of the SCN68000 which are outside the normal processing associated with the execution of instructions. The functions of the bits in the supervisor portion of the status register are covered: the supervisor/user bit, the trace enable bit, and the processor interrupt priority mask. Finally, the sequence of memory references and actions taken by the processor on exception conditions are detailed.

The SCN68000 is always in one of three processing states: normal, exception, or halted. The normal processing state is that associated with instruction execution; the memory references are to fetch instructions and operands, and to store results. A special case of the normal state is the stopped state which the processor enters when a stop instruction is executed. In this state, no further references are made.

The exception processing state is associated with interrupts, trap instructions, tracing, and other exceptional conditions. The exception may be internally generated by an instruction or by an unusual condition arising during the execution of an instruction. Externally, exception processing can be forced by an interrupt, by a bus error, or by a reset. Exception processing is designed to provide an efficient context switch so that the processor may handle unusual conditions.

The halted processing state is an indication of catastrophic hardware failure. For example, if during the exception processing of a bus error another bus error occurs, the processor assumes that the system is unusable and halts. Only an external reset can restart a halted processor. Note that a processor in the stopped state is not in the halted state, nor vice versa.

Privilege States

The processor operates in one of two states of privilege: the "supervisor" state or the "user" state. The privilege state determines which operations are legal, are used to choose between the supervisor stack pointer and the user stack pointer in instruction references, and may be used by an external memory management device to control and translate accesses.

The privilege state is a mechanism for providing security in a computer system. Programs should access only their own code and data areas, and ought to be restricted from accessing information which they do not need and must not modify.

The privilege mechanism provides security by allowing most programs to execute in user state. In this state, the accesses are controlled, and the effects on other parts of the system are limited. The operating system executes in the supervisor state, has access to all resources, and performs the overhead tasks for the user state programs.

Supervisor State

The supervisor state is the higher state of privilege. For instruction execution, the supervisor state is determined by the S bit of the status register; if the S bit is asserted (high), the processor is in the supervisor state. All instructions can be executed in the supervisor state. The bus cycles generated by instructions executed in the supervisor state are classified as supervisor references. While the processor is in the supervisor privilege state, those instructions which use either the system stack pointer implicitly or address register seven explicitly access the supervisor stack pointer.

All exception processing is done in the supervisor state, regardless of the setting of the S bit. The bus cycles generated during exception processing are classified as supervisor references. All stacking operations during exception processing use the supervisor stack pointer.

User State

The user state is the lower state of privilege. For instruction execution, the user state is determined by the S bit of the status register; if the S bit is negated (low), the processor is executing instructions in the user state.

Most instructions execute the same in user state as in the supervisor state. However, some instructions which have important system effects are made privileged. User programs are not permitted to execute the stop instruction or the reset instruction. To ensure that a user program cannot enter the supervisor state except in a controlled manner, the instructions which modify the whole state register are privileged. To aid in debugging programs which are to be used as operating systems, the move to user stack pointer (MOVE to USP) and move from user stack pointer (MOVE from USP) instructions are also privileged.

The bus cycles generated by an instruction executed in the user state are classified as user state references. This allows an external memory management device to translate the address and to control access to protected portions of the address space. While the processor is in the user privilege state, those instructions which use either the system stack pointer implicitly or address register seven explicitly, access the user stack pointer.

Privilege State Changes

Once the processor is in the user state and executing instructions, only exception processing can change the privilege state. During exception processing, the current setting of the S bit of the status register is saved and the S bit is asserted, putting the processor in the supervisor state. Therefore, when instruction execution resumes at the address specified to process the exception, the processor is in the supervisor privilege state.

Table 18. Bus Cycle Classification

FUNCTION CODE OUTPUT			REFERENCE CLASS
FC2	FC1	FC0	
0	0	0	(Unassigned)
0	0	1	User data
0	1	0	User program
0	1	1	(Unassigned)
1	0	0	(Unassigned)
1	0	1	Supervisor data
1	1	0	Supervisor program
1	1	1	Interrupt acknowledge

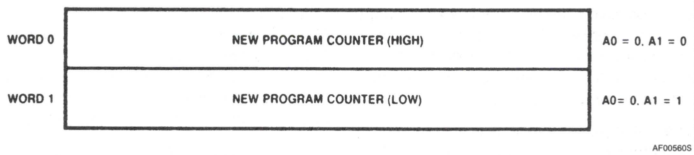
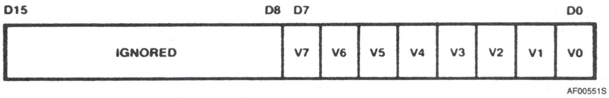


Figure 29. Format of Vector Table Entries



Where:
V7 is the MSB of the vector number.
V8 is the LSB of the vector number.

Figure 30. Vector Number Format

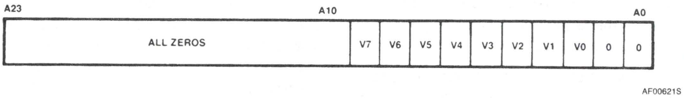


Figure 31. Exception Vector Address Calculation

Reference Classification

When the processor makes a reference, it classifies the kind of reference being made, using the encoding on the three function code output lines. This allows external translation of addresses, control of access, and differentiation of special processor state, such as interrupt acknowledge. Table 18 lists the classification of references.

Exception Processing

Before discussing the details of interrupts, traps, and tracing, a general description of exception processing is in order. The processing of an exception occurs in four steps, with variations for different exception causes. During the first step, a temporary copy of the status register is made and the status register is set for exception processing. In the second step the exception vector is determined and the third step is the saving of the current processor context. In the fourth step a new context is obtained and the processor switches to instruction processing.

Exception Vectors

Exception vectors are memory locations from which the processor fetches the address of a routine which will handle that exception. All exception vectors are two words in length (Figure 29), except for the reset vector which is four words. All exception vectors lie in the supervisor data space, except for the reset vector which is in the supervisor program space. A vector number is an 8-bit number which, when multiplied by four, gives the address of an exception vector. Vector numbers are generated internally or externally, depending on the cause of the exception. In the case of interrupts, during the interrupt acknowledge bus cycle, a peripheral provides an 8-bit vector number (Figure 30) to the processor on data bus lines D0 through D7. The processor translates the vector number into a full 24-bit address, shown in Figure 31. The memory layout for exception vectors is given in Table 19.

Table 19. Exception Vector Table

VECTOR NUMBER(S)	ADDRESS			ASSIGNMENT
	Dec	Hex	Space	
0	0	000	SP	Reset: initial SSP
–	4	004	SP	Reset: initial PC
2	8	008	SD	Bus error
3	12	00C	SD	Address error
4	16	010	SD	Illegal instruction
5	20	014	SD	Zero divide
6	24	018	SD	CHK instruction
7	28	01C	SD	TRAPV instruction
8	32	020	SD	Privilege violation
9	36	024	SD	Trace
10	40	028	SD	Line 1010 emulator
11	44	02C	SD	Line 1111 emulator
12*	48	030	SD	(Unassigned, reserved)
13*	52	034	SD	(Unassigned, reserved)
14*	56	038	SD	(Unassigned, reserved)
15	60	03C	SD	Uninitialized interrupt vector
16 – 23*	64	04C	SD	(Unassigned, reserved)
	95	05F		–
24	96	060	SD	Spurious interrupt
25	100	064	SD	Level 1 interrupt autovector
26	104	068	SD	Level 2 interrupt autovector
27	108	06C	SD	Level 3 interrupt autovector
28	112	070	SD	Level 4 interrupt autovector
29	116	074	SD	Level 5 interrupt autovector
30	120	078	SD	Level 6 interrupt autovector
31	124	07C	SD	Level 7 interrupt autovector
32 – 47	128	080	SD	TRAP instruction vectors
	191	0BF		–
48 – 63*	192	0C0	SD	(Unassigned, reserved)
	255	0FF		–
64 – 255	256	100	SD	User interrupt vectors
	1023	3FF		–

NOTE:
*Vector numbers 12, 13, 14, 16 through 23, and 48 through 63 are reserved for future enhancements by Signetics. No user peripheral devices should be assigned these numbers.

As shown in Table 19, the memory layout is 512 words long (1024 bytes). It starts at address 0 and proceeds through address 1023. This provides 255 unique vectors; some of these are reserved for TRAPs and other system functions. Of the 255, there are 192 reserved for user interrupt vectors. However, there is no protection on the first 64 entries, so user interrupt vectors may overlap at the discretion of the systems designer.

Kinds of Exceptions

Exceptions can be generated by either internal or external causes. The externally generated exceptions are the interrupts and the bus error and reset requests. The interrupts are requests from peripheral devices for processor action while the bus error and reset inputs are used for access control and processor restart. The internally generated exceptions come from instructions, or from address errors or tracing. The trap (TRAP), trap on overflow (TRAPV), check data register against upper bounds (CHK), and divide (DIV) instructions all can generate exceptions as part of their instruction execution. In addition, illegal instructions, word fetches from odd addresses, and privilege violations cause exceptions. Tracing behaves like a very high-priority internally-generated interrupt after each instruction execution.

Exception Processing Sequence

Exception processing occurs in four identifiable steps. In the first step, an internal copy is made of the status register. After the copy is made, the S bit is asserted, putting the processor into the supervisor privilege state. Also, the T bit is negated which will allow the exception handler to execute unhindered by tracing. For the reset and interrupt exceptions, the interrupt priority mask is also updated.

In the second step, the vector number of the exception is determined. For interrupts, the vector number is obtained by a processor fetch and classified as an interrupt acknowledge. For all other exceptions, internal logic provides the vector number. This vector number is then used to generate the address of the exception vector.

The third step is to save the current processor status, except for the reset exception. The current program counter value and the saved copy of the status register are stacked using the supervisor stack pointer as shown in Figure 32. The program counter value stacked usually points to the next unexecuted instruction; however, for bus error and address error, the value stacked for the program counter is unpredictable, and may be incremented from the address of the instruction which caused the error. Additional information defining the current context is stacked for the bus error and address error exceptions.

The last step is the same for all exceptions. The new program counter value is fetched from the exception vector. The processor then resumes instruction execution. The instruction at the address given in the exception vector is fetched, and normal instruction decoding and execution is started.

Multiple Exceptions

These paragraphs describe the processing which occurs when multiple exceptions arise simultaneously. Exceptions can be grouped according to their occurrence and priority. The group 0 exceptions are reset, bus error, and address error. These exceptions cause the instruction currently being executed to be aborted and the exception processing to commence within two clock cycles.

The group 1 exceptions are trace and interrupt, as well as the privilege violations and

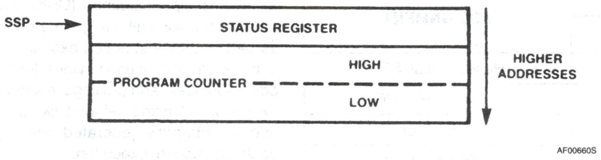


Figure 32. Exception Stack Order (Groups 1 and 2)

Table 20. Exception Grouping and Priority

GROUP	EXCEPTION	PROCESSING
0	Reset address error bus error	Exception processing begins within two clock cycles
1	Trace interrupt illegal privilege	Exception processing begins before the next instruction
2	TRAP, TRAPV, CHK, zero divide	Exception processing is started by normal instruction execution

illegal instructions. These exceptions allow the current instruction to execute to completion, but pre-empt the execution of the next instruction by forcing exception processing to occur (privilege violations and illegal instructions are detected when they are the next instruction to be executed). The group 2 exceptions occur as part of the normal processing of instructions. The TRAP, TRAPV, CHK, and zero divide exceptions are in this group. For these exceptions, the normal execution of an instruction may lead to exception processing.

Group 0 exceptions have highest priority, while group 2 exceptions have lowest priority. Within group 0, reset has highest priority, followed by bus error and then address error. Within group 1, trace has priority over external interrupts, which in turn takes priority over illegal instruction and privilege violation. Since only one instruction can be executed at a time, there is no priority relation within group 2.

The priority relation between two exceptions determines which is taken, or taken first, if the conditions for both arise simultaneously. Therefore, if a bus error occurs during a TRAP instruction, the bus error takes precedence, and the TRAP instruction processing is aborted. In another example, if an interrupt request occurs during the execution of an instruction while the T bit is asserted, the trace exception has priority, and is processed first. Before instruction processing resumes, however, the interrupt exception is also processed, and instruction processing commences finally in the interrupt handler routine. A summary of exception grouping and priority is given in Table 20.

Exception Processing Detailed Discussion

Exceptions have a number of sources and each exception has processing which is peculiar to it. The following paragraphs detail the sources of exceptions, how each arises, and how each is processed.

Reset

The reset input provides the highest exception level. The processing of the reset signal is designed for system initiation and recovery from catastrophic failure. Any processing in progress at the time of the reset is aborted and cannot be recovered. The processor is forced into the supervisor state and the trace state is forced off. The processor interrupt priority mask is set at level seven. The vector number is internally generated to reference the reset exception vector at location 0 in the supervisor program space. Because no assumptions can be made about the validity of register contents, in particular the supervisor stack pointer, neither the program counter nor the status register is saved. The address contained in the first two words of the reset exception vector is fetched as the initial supervisor stack pointer, and the address in the last two words of the reset exception vector is fetched as the initial program counter. Finally, instruction execution is started at the address in the program counter. The power-up/restart code should be pointed to by the initial program counter.

The reset instruction does not cause loading of the reset vector, but does assert the reset line to reset external devices. This allows the software to reset the system to a known state and then continue processing with the next instruction.

Interrupts

Seven levels of interrupt priorities are provided. Devices may be chained externally within interrupt priority levels, allowing an unlimited number of peripheral devices to interrupt the processor. Interrupt priority levels are numbered from one to seven, with level seven being the highest priority. The status register contains a 3-bit mask which indicates the current processor priority, and interrupts are inhibited for all priority levels less than or equal to the current processor priority.

An interrupt request is made to the processor by encoding the interrupt request level on the interrupt request lines; a zero indicates no interrupt request. Interrupt requests arriving at the processor do not force immediate exception processing, but are made pending. Pending interrupts are detected between instruction executions. If the priority of the pending interrupt is lower than or equal to the current processor priority, execution continues with the next instruction and the interrupt exception processing is postponed. (The recognition of level seven is slightly different, as explained in the following paragraph.)

If the priority of the pending interrupt is greater than the current processor priority, the exception processing sequence is started. A copy of the status register is saved, the privilege state is sent to the supervisor stack, tracing is suppressed, and the processor priority level is set to the level of the interrupt acknowledged. The processor fetches the vector number from the interrupting device, classifying the reference as an interrupt acknowledge and displaying the level number of the interrupt being acknowledged on the address bus. If external logic requests an automatic vectoring, the processor internally generates a vector number which is determined by the interrupt level number. If external logic indicates a bus error, the interrupt is taken to be spurious, and the generated vector number references the spurious interrupt vector. The processor then proceeds with the usual exception processing, saving the program counter and status register on the supervisor stack. The saved value of the program counter is the address of the instruction which would have been executed had the interrupt not been present. The content of the interrupt vector whose vector number was previously obtained is fetched and loaded into the program counter, and normal instruction execution commences in the interrupt handling routine. A flowchart for the interrupt acknowledge sequence is given in Figure 33, a timing diagram is given in Figure 34, and the interrupt processing sequence is shown in Figure 35.

Priority level seven is a special case. Level seven interrupts cannot be inhibited by the interrupt priority mask, thus providing a "non-maskable interrupt" capability. An interrupt is generated each time the interrupt request level changes from some lower level to level seven. Note that a level seven interrupt may still be caused by the level comparison if the request level is a seven and the processor priority is set to a lower level by an instruction.

Uninitialized Interrupt

An interrupting device asserts $\overline{\text{VPA}}$ or provides an interrupt during an interrupt acknowledge cycle to the SCN68000. If the vector register has not been initialized, the responding S68000 family peripheral will provide vector 15, the uninitialized interrupt vector. This provides a uniform way to recover from a programming error.

Spurious Interrupt

If during the interrupt acknowledge cycle no device responds by asserting $\overline{\text{DTACK}}$ or $\overline{\text{VPA}}$, the bus error line should be asserted to terminate the vector acquisition. The processor separates the processing of this error from bus error by fetching the spurious interrupt vector instead of the bus error vector. The processor then proceeds with the usual exception processing.

Instruction Traps

Traps are exceptions caused by instructions. They arise either from processor recognition of abnormal conditions during instruction execution, or from use of instructions whose normal behavior is trapping.

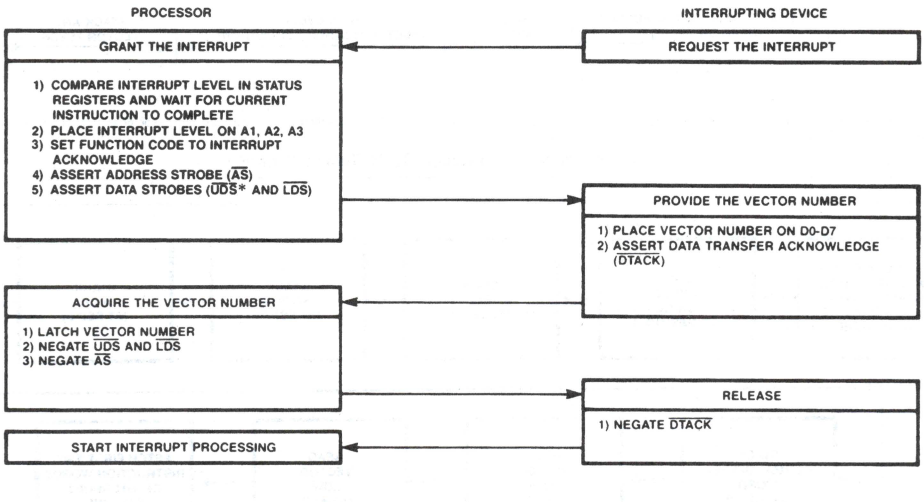
Some instructions are used specifically to generate traps. The TRAP instruction always forces an exception and is useful for implementing system calls for user programs. The TRAPV and CHK instructions force an exception if the user program detects a runtime error, which may be an arithmetic overflow or a subscript out of bounds.

The signed divide (DIVS) and unsigned (DIVU) instructions will force an exception if a division operation is attempted with a divisor of zero.

Illegal and Unimplemented Instructions

"Illegal instruction" is the term used to refer to any of the word bit patterns which are not the bit pattern of the first word of a legal instruction. During instruction execution, if such an instruction is fetched, an illegal instruction exception occurs. Signetics reserves the right to define instructions whose opcodes may be any of the illegal instructions. Three bit patterns will always force an illegal instruction trap on all S68000 family compatible microprocessors. They are: $4AFA, $4AFB, and $4AFC. Two of the patterns, $4AFA and $4AFB, are reserved for Signetics systems products. The third pattern, $4AFC, is reserved for customer use.

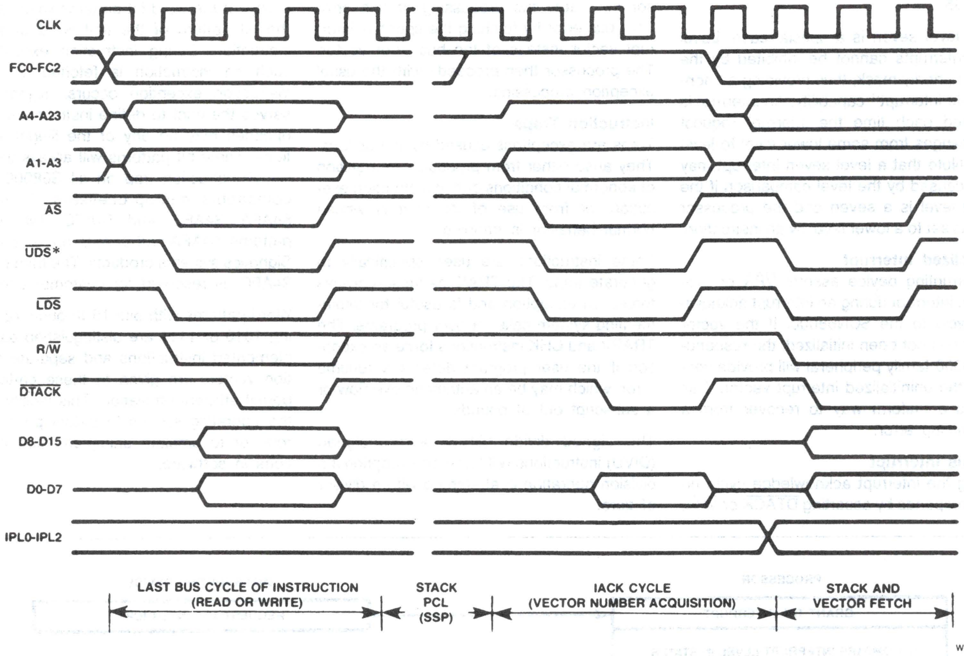
Word patterns with bits 15 through 12 equaling 1010 or 1111 are distinguished as unimplemented instructions and separate exception vectors are given to these patterns to permit efficient emulation. This facility allows the operating system to detect program errors, or to emulate unimplemented instructions in software.


PROCESSOR
INTERRUPTING DEVICE
GRANT THE INTERRUPT
REQUEST THE INTERRUPT
1) COMPARE INTERRUPT LEVEL IN STATUS REGISTERS AND WAIT FOR CURRENT INSTRUCTION TO COMPLETE
2) PLACE INTERRUPT LEVEL ON A1, A2, A3
3) SET FUNCTION CODE TO INTERRUPT ACKNOWLEDGE
4) ASSERT ADDRESS STROBE (AS)
5) ASSERT DATA STROBES (UDS* AND LDS)
PROVIDE THE VECTOR NUMBER
1) PLACE VECTOR NUMBER ON D0-D7
2) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
ACQUIRE THE VECTOR NUMBER
1) LATCH VECTOR NUMBER
2) NEGATE UDS AND LDS
3) NEGATE AS
RELEASE
1) NEGATE DTACK
START INTERRUPT PROCESSING
AF00652S

NOTE:

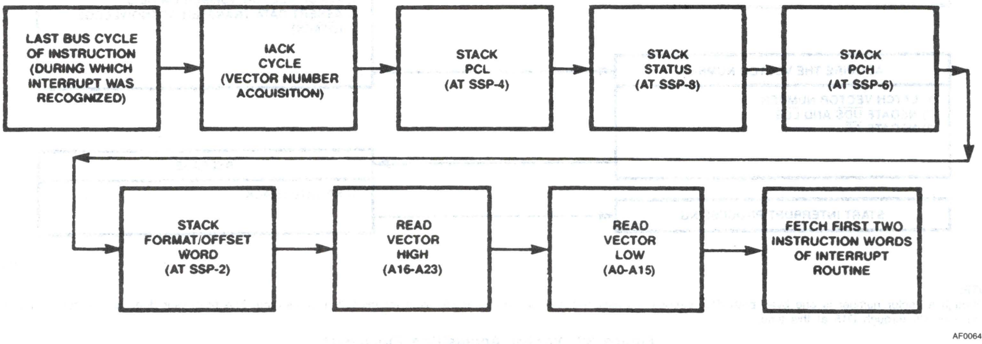
*Although a vector number is one byte, both data strobes are asserted due to the microcode used for exception processing. The processor does not recognize anything on data lines D8 through D15 at this time.

Figure 33. Vector Acquisition Flowchart


CLK
FC0-FC2
A4-A23
A1-A3
AS
UDS*
LDS
R/W
DTACK
D8-D15
D0-D7
IPL0-IPL2
LAST BUS CYCLE OF INSTRUCTION (READ OR WRITE)
STACK PCL (SSP)
IACK CYCLE (VECTOR NUMBER ACQUISITION)
STACK AND VECTOR FETCH
WF01531S

NOTE:
*Although a vector number is one byte, both data strobes are asserted due to the microcode used for exception processing. The processor does not recognize anything on data lines D8 through D15 at this time.

Figure 34. Interrupt Acknowledge Cycle Timing Diagram


LAST BUS CYCLE OF INSTRUCTION (DURING WHICH INTERRUPT WAS RECOGNIZED)
IACK CYCLE (VECTOR NUMBER ACQUISITION)
STACK PCL (AT SSP-4)
STACK STATUS (AT SSP-8)
STACK PCH (AT SSP-6)
STACK FORMAT/OFFSET WORD (AT SSP-2)
READ VECTOR HIGH (A16-A23)
READ VECTOR LOW (A0-A15)
FETCH FIRST TWO INSTRUCTION WORDS OF INTERRUPT ROUTINE
AF00640S

NOTE:
SSP refers to the value of the supervisor stack pointer before the interrupt occurs.

Figure 35. Interrupt Processing Sequence

Privilege Violations

In order to provide system security, various instructions are privileged. An attempt to execute one of the privileged instructions while in the user state will cause an exception. The privileged instructions are:

STOP
RESET
RTE
MOVE to SR
AND Immediate to SR
EOR Immediate to SR
OR Immediate to SR
MOVE to USP

Tracing

To aid in program development, the SCN68000 includes a facility to allow instruction-by-instruction tracing. In the trace state, after each instruction is executed an exception is forced, allowing a debugging program to monitor the execution of the program under test.

The trace facility uses the T bit in the supervisor portion of the status register. If the T bit is negated (off), tracing is disabled, and instruction execution proceeds from instruction to instruction as normal. If the T bit is asserted (on) at the beginning of the execution of an instruction, a trace exception will be generated after the execution of that instruction is completed. If the instruction is not executed, either because an interrupt is taken, or the instruction is illegal or privileged, the trace exception does not occur. The trace exception also does not occur if the instruction is aborted by a reset, bus error, or address error exception. If the instruction is indeed executed and an interrupt is pending on completion, the trace exception is processed before the interrupt exception. If, during the execution of the instruction an exception is forced by that instruction, the forced exception is processed before the trace exception.

As an extreme illustration of the above rules, consider the arrival of an interrupt during the execution of a TRAP instruction while tracing is enabled. First the trap exception is processed, then the trace exception, and finally the interrupt exception. Instruction execution resumes in the interrupt handler routine.

Bus Error

Bus error exceptions occur when the external logic requests that a bus error be processed by an exception. The current bus cycle which the processor is making is then aborted. Whether the processor was doing instruction or exception processing, that processing is terminated, and the processor immediately begins exception processing.

Exception processing for the bus error follows the usual sequence of steps. The status register is copied, the supervisor state is entered, and the trace state is turned off. The vector number is generated to refer to the bus error vector. Since the processor was not between instructions when the bus error exception request was made, the context of the processor is more detailed. To save more of this context, additional information is saved on the supervisor stack. The program counter and the copy of the status register are of course saved. The value saved for the program counter is advanced by some amount, one to five words beyond the address of the first word of the instruction which made the reference causing the bus error. If the bus error occurred during the fetch of the next instruction, the saved program counter has a value in the vicinity of the current instruction, even if the current instruction is a branch, a jump, or a return instruction. Besides the usual information, the processor saves its internal copy of the first word of the instruction being processed and the address which was being accessed by the aborted bus cycle. Specific information about the access is also saved: whether it was a read or a write, whether or not the processor was processing an instruction, and the classification displayed on the function code outputs when the bus error occurred. The processor is processing an instruction if it is in the normal state or processing a group 2 exception; the processor is not processing an instruction if it is processing a group 0 or a group 1 exception. Figure 36 illustrates how this information is organized on the supervisor stack. Although this information is not sufficient in general to effect full recovery from the bus error, it does allow software diagnosis. Finally, the processor commences instruction processing at the address contained in vector number two. It is the responsibility of the error handler routine to clean up the stack and determine where to continue execution.

If a bus error occurs during the exception processing for a bus error, address error, or reset, the processor is halted and all processing ceases. This simplifies the detection of catastrophic system failure, since the processor removes itself from the system rather than destroy any memory contents. Only the $\overline{\text{RESET}}$ pin can restart a halted processor.

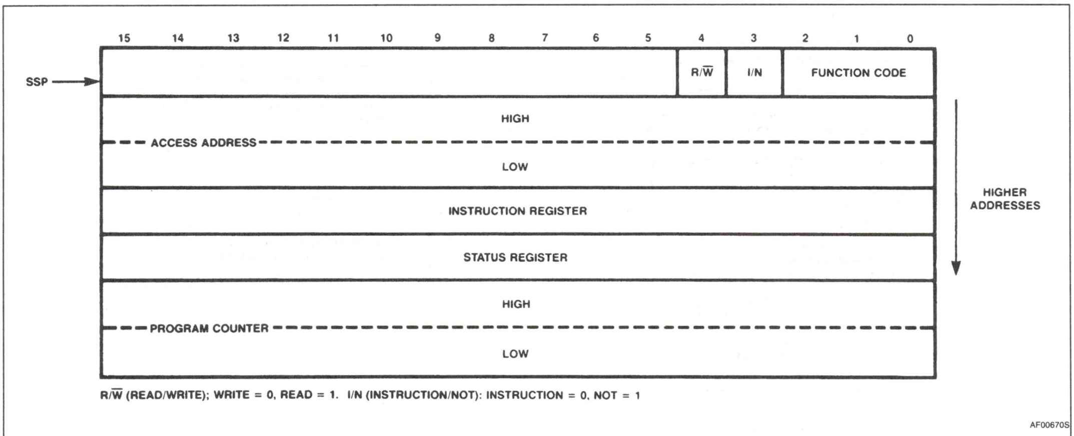


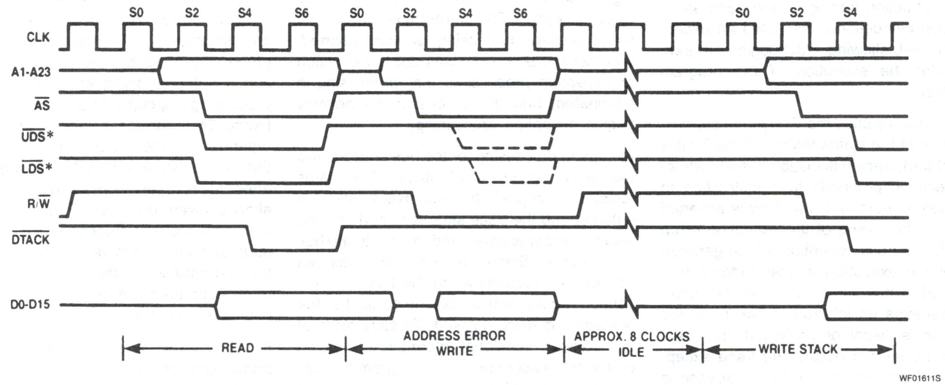
Figure 36. Exception Stack Order (Group 0)

Address Error

Address error exceptions occur when the processor attempts to access a word or a long word operand or an instruction at an odd address. The effect is much like an internally generated bus error, so that the bus cycle is aborted and the processor ceases whatever processing it is currently doing and begins exception processing. After the exception processing commences, the sequence is the same as that for bus error including the information that is stacked, except that the vector number refers to the address error vector instead. Likewise, if an address error occurs during the exception processing for a bus error, address error, or reset, the processor is halted. As shown in Figure 37, an address error will execute a short bus cycle followed by exception processing.

INTERFACE WITH SYNCHRONOUS PERIPHERALS

To interface the synchronous peripherals with the asynchronous SCN68000, the processor modifies its bus cycle to meet the synchronous cycle requirements whenever a synchronous device address is detected. This is possible since both processors use memory mapped I/O. Figure 38 is a flowchart of the interface operation between the processor and synchronous devices.


S0
S2
S4
S6
S0
S2
S4
S6
S0
S2
S4
CLK
A1-A23
$\overline{AS}$
$\overline{UDS}$*
$\overline{LDS}$*
R/$\overline{W}$
$\overline{DTACK}$
D0-D15
READ
ADDRESS ERROR WRITE
APPROX. 8 CLOCKS IDLE
WRITE STACK
WF01611S

NOTE:
*$\overline{UDS}$ and $\overline{LDS}$ are asserted on mask sets R9M, BF4, T6E, DL6, CC1, and GN7.

Figure 37. Address Error Timing Diagram

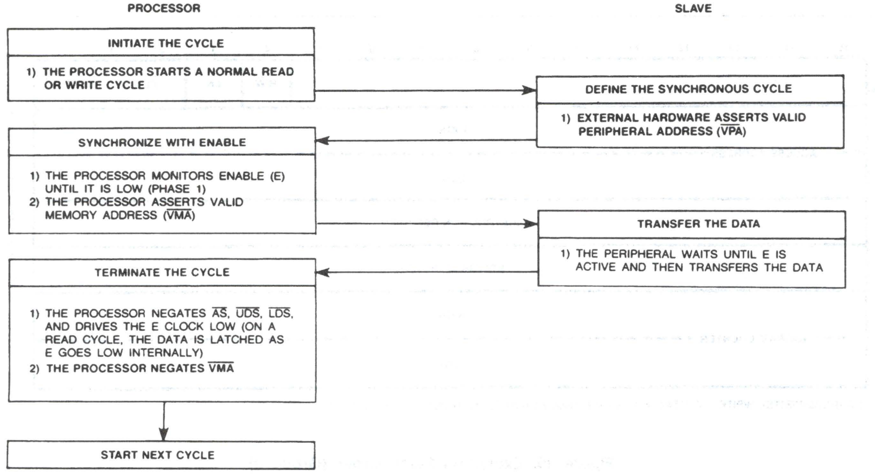

PROCESSOR
SLAVE
INITIATE THE CYCLE
1) THE PROCESSOR STARTS A NORMAL READ OR WRITE CYCLE
DEFINE THE SYNCHRONOUS CYCLE
1) EXTERNAL HARDWARE ASSERTS VALID PERIPHERAL ADDRESS ($\overline{VPA}$)
SYNCHRONIZE WITH ENABLE
1) THE PROCESSOR MONITORS ENABLE (E) UNTIL IT IS LOW (PHASE 1)
2) THE PROCESSOR ASSERTS VALID MEMORY ADDRESS ($\overline{VMA}$)
TRANSFER THE DATA
1) THE PERIPHERAL WAITS UNTIL E IS ACTIVE AND THEN TRANSFERS THE DATA
TERMINATE THE CYCLE
1) THE PROCESSOR NEGATES $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, AND DRIVES THE E CLOCK LOW (ON A READ CYCLE, THE DATA IS LATCHED AS E GOES LOW INTERNALLY)
2) THE PROCESSOR NEGATES $\overline{VMA}$
START NEXT CYCLE
PF00301S

Figure 38. Synchronous Interfacing Flowchart

Data Transfer Operation

Three signals on the processor provide the synchronous interface. They are: enable (E), valid memory address ($\overline{\text{VMA}}$), and valid peripheral address ($\overline{\text{VPA}}$). Enable corresponds to the E or phase 2 signal in existing synchronous systems. The bus frequency is one tenth of the incoming SCN68000 clock frequency. The timing of E allows 1MHz peripherals to be used with 8MHz SCN68000s. Enable has a 60/40 duty cycle; that is, it is low for six input clocks and high for four input clocks. This duty cycle allows the processor to do successive $\overline{\text{VPA}}$ accesses on successive E pulses.

Synchronous cycle timing is given in Figures 39, 40, 48, and 49. At state zero (S0) in the cycle, the address bus is in the high-impedance state. A function code is asserted on the function code output lines. One-half clock later, in state 1, the address bus is released from the high-impedance state.

During state 2, the address strobe ($\overline{\text{AS}}$) is asserted to indicate that there is a valid address on the address bus. If the bus cycle is a read cycle, the upper and/or lower data strobes are also asserted in state 2. If the bus cycle is a write cycle, the read/write (R/$\overline{\text{W}}$) signal is switched to low (write) during state 2. One-half clock later, in state 3, the write data is placed on the data bus, and in state 4 the data strobes are issued to indicate valid data on the data bus. The processor now inserts wait states until it recognizes the assertion of $\overline{\text{VPA}}$.

The $\overline{\text{VPA}}$ input signals the processor that the address on the bus is the address of a synchronous device (or an area reserved for synchronous devices) and that the bus should conform to the phase 2 transfer characteristics of the synchronous bus. Valid peripheral address is derived by decoding the address bus, conditioned by the address strobe. Chip select for the synchronous peripherals should be derived by decoding the address bus conditioned by $\overline{\text{VMA}}$.

After recognition of $\overline{\text{VPA}}$, the processor assures that the enable (E) is low, by waiting if necessary, and subsequently asserts $\overline{\text{VMA}}$. Valid memory address is then used as part of the chip select equation of the peripheral. This ensures that the synchronous peripherals are selected and deselected at the correct time. The peripheral now runs its cycle during the high portion of the E signal. Figures 39 and 40 depict the best and worst case synchronous cycle timing. This cycle length is dependent strictly on when $\overline{\text{VPA}}$ is asserted in relationship to the E clock.

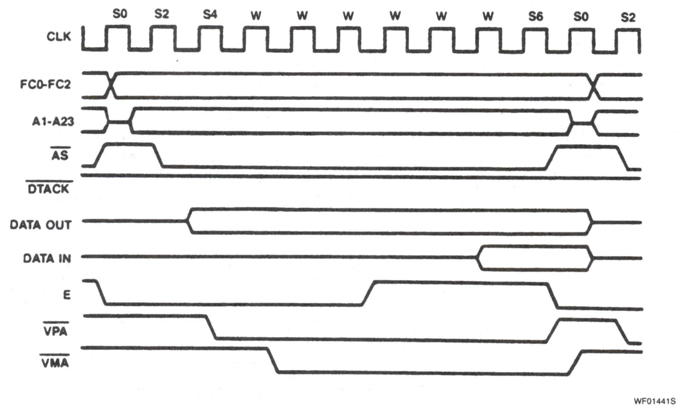


Figure 39. SCN68000 to Synchronous Peripheral Timing — Best Case

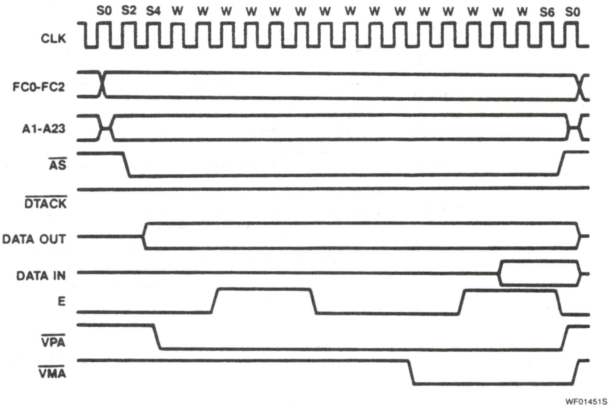


Figure 40. SCN68000 to Synchronous Peripheral Timing—Worst Case

If we assume that external circuitry asserts $\overline{\text{VPA}}$ as soon as possible after the assertion of $\overline{\text{AS}}$, then $\overline{\text{VPA}}$ will be recognized as being asserted on the falling edge of S4. In this case, no "extra" wait cycles will be inserted prior to the recognition of $\overline{\text{VPA}}$ asserted and only the wait cycles inserted to synchronize with the E clock will determine the total length of the cycle. In any case, the synchronization delay will be some integral number of clock cycles within the following two extremes:

1. Best Case — $\overline{\text{VPA}}$ is recognized as being asserted on the falling edge three clock cycles before E rises (or three clock cycles after E falls).
2. Worst Case — $\overline{\text{VPA}}$ is recognized as being asserted on the falling edge two clock cycles before E rises (or four clock cycles after E falls).

During a read cycle, the processor latches the peripheral data in state 6. For all cycles, the processor negates the address and data strobes one-half clock cycle later in state 7 and the enable signal goes low at this time. Another half clock later, the address bus is put in the high-impedance state. During a write cycle, the data bus is put in the high-impedance state and the read/write signal is switched high. The peripheral logic must remove $\overline{\text{VPA}}$ within one clock after the address strobe is negated.

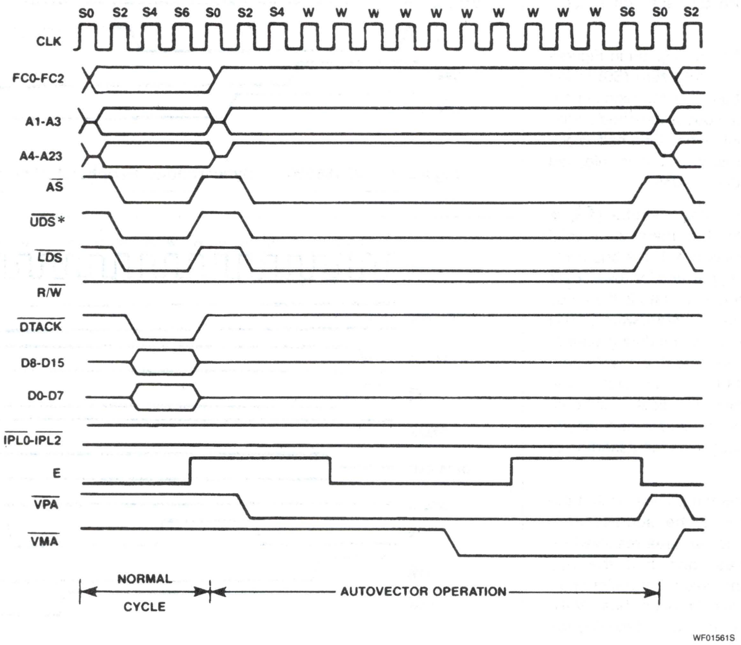
$\overline{\text{DTACK}}$ should not be asserted while $\overline{\text{VPA}}$ is asserted. Notice that the SCN68000 $\overline{\text{VMA}}$ is active low, contrasted with the active high synchronous $\overline{\text{VMA}}$. This allows the processor to put its buses in the high-impedance state on DMA requests without inadvertently selecting the peripherals.

Interrupt Interface Operation

During an interrupt acknowledge cycle while the processor is fetching the vector, the $\overline{\text{VPA}}$ is asserted, the SCN68000 will assert $\overline{\text{VMA}}$ and complete a normal synchronous read cycle as shown in Figure 41. The processor will then use an internally generated vector that is a function of the interrupt being serviced. This process is known as autovectoring. The seven autovectors are vector numbers 25 through 31 (decimal).

Autovectoring operates in the same fashion (but is not restricted to) the synchronous interrupt sequence. The basic difference is that there are six normal interrupt vectors and one NMI type vector. As with both the synchronous and the SCN68000's normal vectored interrupt, the interrupt service routine can be located anywhere in the address space. This is due to the fact that while the vector numbers are fixed, the contents of the vector table entries are assigned by the user.

Since $\overline{\text{VMA}}$ is asserted during autovectoring, care should be taken to insure the synchronous peripheral address decoding prevents unintended accesses.



NOTE:
*Although $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$ are asserted, no data is read from the bus during the autovector cycle. The vector number is generated internally.

Figure 41. Autovector Operation Timing Diagram

INSTRUCTION SET AND EXECUTION TIMES

Instruction Set

The following paragraphs provide information about the addressing categories and instruction set of the SCN68000.

Addressing Categories

Effective address modes may be categorized by the ways in which they may be used. The following classifications will be used in the instruction definitions.

These categories may be combined, so that additional, more restrictive, classifications may be defined. For example, the instruction descriptions use such classifications as alterable memory or data alterable. The former refers to those addressing modes which are both alterable and memory addresses, and the latter refers to addressing modes which are both data and alterable.

Table 21 shows the various categories to which each of the effective address modes belong. Table 22 is the instruction set summary.

Data	If an effective address mode may be used to refer to data operands, it is considered a data addressing effective address mode.
Memory	If an effective address mode may be used to refer to memory operands, it is considered a memory addressing effective address mode.
Alterable	If an effective address mode may be used to refer to alterable (writable) operands, it is considered an alterable addressing effective address mode.
Control	If an effective address mode may be used to refer to memory operands without an associated size, it is considered a control addressing effective address mode.

Table 21. Effective Addressing Mode Categories

EFFECTIVE ADDRESS MODES	MODE	REGISTER	ADDRESSING CATEGORIES			
			Data	Memory	Control	Alterable
Dn	000	Register number	X	–	–	X
An	001	Register number	–	–	–	X
(An)	010	Register number	X	X	X	X
(An)+	011	Register number	X	X	–	X
–(An)	100	Register number	X	X	–	X
d(An)	101	Register number	X	X	X	X
d(An, ix)	110	Register number	X	X	X	X
xxx.W	111	000	X	X	X	X
xxx.L	111	001	X	X	X	X
d(PC)	111	010	X	X	X	–
d(PC, ix)	111	011	X	X	X	–
#xxx	111	100	X	X	–	–

Table 22. Instruction Set

MNEMONIC	DESCRIPTION	OPERATION	CONDITION CODES				
			X	N	Z	V	C
ABCD	Add decimal with extend	(Destination)$_{10}$ + (Source)$_{10}$ + X → Destination	*	U	*	U	*
ADD	Add binary	(Destination) + (Source) → Destination	*	*	*	*	*
ADDA	Add address	(Destination) + (Source) → Destination	–	–	–	–	–
ADDI	Add immediate	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDQ	Add quick	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDX	Add extended	(Destination) + (Source) + X → Destination	*	*	*	*	*
AND	AND logical	(Destination) $\wedge$ (Source) → Destination	–	*	*	0	0
ANDI	AND immediate	(Destination) $\wedge$ Immediate Data → Destination	–	*	*	0	0
ANDI to CCR	AND immediate to condition codes	(Source) $\wedge$ CCR → CCR	*	*	*	*	*
ANDI to SR	AND immediate to status register	(Source) $\wedge$ SR → SR	*	*	*	*	*
ASL, ASR	Arithmetic shift	(Destination) Shifted by < count > → Destination	*	*	*	*	*
B_{CC}	Branch conditionally	If CC then PC + d → PC	–	–	–	–	–
BCHG	Test a bit and change	~(< bit number >) OF Destination → Z ~(< bit number >) OF Destination → < bit number > OF Destination	–	–	*	–	–
BCLR	Test a bit and clear	~(< bit number >) OF Destination → Z 0 → < bit number > → OF Destination	–	–	*	–	–
BRA	Branch always	PC + d → PC	–	–	–	–	–
BSET	Test a bit and set	~(< bit number >) OF Destination → Z 1 → < bit number > OF Destination	–	–	*	–	–
BSR	Branch to subroutine	PC → –(SP); PC + d → PC	–	–	–	–	–
BTST	Test a bit	~(< bit number >) OF Destination → Z	–	–	*	–	–
CHK	Check register against bounds	If Dn < 0 or Dn > (< ea >) then TRAP	–	*	U	U	U
CLR	Clear an operand	0 → Destination	–	0	1	0	0
CMP	Compare	(Destination) – (Source)	–	*	*	*	*
CMPA	Compare address	(Destination) – (Source)	–	*	*	*	*
CMPI	Compare immediate	(Destination) – Immediate Data	–	*	*	*	*
CMPM	Compare memory	(Destination) – (Source)	–	*	*	*	*
DB_{CC}	Test condition, decrement and branch	if ~CC then Dn – 1 → Dn; if Dn ≠ –1 then PC + d → PC	–	–	–	–	–
DIVS	Signed divide	(Destination)/(Source) → Destination	–	*	*	*	0
DIVU	Unsigned divide	(Destination)/(Source) → Destination	–	*	*	*	0
EOR	Exclusive OR logical	(Destination) ⊕ (Source) → Destination	–	*	*	0	0
EORI	Exclusive OR immediate	(Destination) ⊕ Immediate Data → Destination	–	*	*	0	0
EORI to CCR	Exclusive OR immediate to condition codes	(Source) ⊕ CCR → CCR	*	*	*	*	*
EORI to SR	Exclusive OR immediate to status register	(Source) ⊕ SR → SR	*	*	*	*	*
EXG	Exchange register	Rx ↔ Ry	–	–	–	–	–
EXT	Sign extend	(Destination) Sign-extended → Destination	–	*	*	0	0
JMP	Jump	Destination → PC	–	–	–	–	–
JSR	Jump to subroutine	PC → –(SP); Destination → PC	–	–	–	–	–
LEA	Load effective address	Destination → An	–	–	–	–	–
LINK	Link and allocate	An → –(SP); SP → An; SP + Displacement → SP	–	–	–	–	–
LSL, LSR	Logical shift	(Destination) Shifted by < count > → Destination	*	*	*	0	*
MOVE	Move data from source to destination	(Source) → Destination	–	*	*	0	0
MOVE to CCR	Move to condition code	(Source) → CCR	*	*	*	*	*
MOVE to SR	Move to status register	(Source) → SR	*	*	*	*	*
MOVE from SR	Move from status register	SR → Destination	–	–	–	–	–

Table 22. Instruction Set (Continued)

MNEMONIC	DESCRIPTION	OPERATION	CONDITION CODES				
			X	N	Z	V	C
MOVE USP	Move user stack pointer	USP → An; An → USP	–	–	–	–	–
MOVEA	Move address	(Source) → Destination	–	–	–	–	–
MOVEM	Move multiple registers	Registers → Destination (Source) → Registers	–	–	–	–	–
MOVEP	Move peripheral data	(Source) → Destination	–	–	–	–	–
MOVEQ	Move quick	Immediate Data → Destination	–	*	*	0	0
MULS	Signed multiply	(Destination) X (Source) → Destination	–	*	*	0	0
MULU	Unsigned multiply	(Destination) X (Source) → Destination	–	*	*	0	0
NBCD	Negate decimal with extend	0 – (Destination)$_{10}$ – X → Destination	*	U	*	U	*
NEG	Negate	0 – (Destination) → Destination	*	*	*	*	*
NEGX	Negate with extend	0 – (Destination) – X → Destination	*	*	*	*	*
NOP	No operation	–	–	–	–	–	–
NOT	Logical complement	~ (Destination) → Destination	–	*	*	0	0
OR	Inclusive OR logical	(Destination) v (Source) → Destination	–	*	*	0	0
ORI	Inclusive OR immediate	(Destination) v Immediate Data → Destination	–	*	*	0	0
ORI to CCR	Inclusive OR immediate to conditions codes	(Source) v CCR → CCR	*	*	*	*	*
ORI to SR	Inclusive OR immediate to status register	(Source) v SR → SR	*	*	*	*	*
PEA	Push effective address	Destination → – (SP)	–	–	–	–	–
RESET	Reset external devices	–	–	–	–	–	–
ROL, ROR	Rotate (without extend)	(Destination) Rotated by < count > → Destination	–	*	*	0	*
ROXL, ROXR	Rotate with extend	(Destination) Rotated by < count > → Destination	*	*	*	0	*
RTE	Return from exception	(SP) + → SR; (SP) + → PC	*	*	*	*	*
RTR	Return and restore condition codes	(SP) + → CC; (SP) + → PC	*	*	*	*	*
RTS	Return from subroutine	(SP) + → PC	–	–	–	–	–
SBCD	Subtract decimal with extend	(Destination)$_{10}$ – (Source)$_{10}$ – X → Destination	*	U	*	U	*
S$_{CC}$	Set according to condition	If CC then 1's → Destination else 0's → Destination	–	–	–	–	–
STOP	Load status register and stop	Immediate Data → SR; STOP	*	*	*	*	*
SUB	Subtract binary	(Destination) – (Source) → Destination	*	*	*	*	*
SUBA	Subtract address	(Destination) – (Source) → Destination	–	–	–	–	–
SUBI	Subtract immediate	(Destination) – Immediate Data → Destination	*	*	*	*	*
SUBQ	Subtract quick	(Destination) – Immediate Data → Destination)	*	*	*	*	*
SUBX	Subtract with extend	(Destination) – (Source) – X → Destination	*	*	*	*	*
SWAP	Swap register halves	Register [31:16] ↔ Register [15:0]	–	*	*	0	0
TAS	Test and set an operand	(Destination) Tested → CC; 1 → [7] OF Destination	–	*	*	0	0
TRAP	Trap	PC → – (SSP); SR → – (SSP); (Vector) → PC	–	–	–	–	–
TRAPV	Trap on overflow	If V then TRAP	–	–	–	–	–
TST	Test and operand	(Destination) Tested → CC	–	*	*	0	0
UNLK	Unlink	An → SP; (SP) + → An	–	–	–	–	–

NOTES:

[ ] = bit number
⊕ logical exclusive OR
∧ logical AND
V logical OR
~ logical complement

* affected
– unaffected
0 cleared
1 set
U undefined

Instruction Prefetch

The SCN68000 uses a two-word tightly-coupled instruction prefetch mechanism to enhance performance. This mechanism is described in terms of the microcode operations involved. If the execution of an instruction is defined to begin when the microroutine for that instruction is entered, some features of the prefetch mechanism can be described.

1. When execution of an instruction begins, the operation word and the word following have already been fetched. The operation word is in the instruction decoder.
2. In the case of multi-word instructions, as each additional word of the instruction is used internally, a fetch is made to the instruction stream to replace it.
3. The last fetch for an instruction from the instruction stream is made when the operation word is discarded and decoding is started on the next instruction.
4. If the instruction is a single-word instruction causing a branch, the second word is not used. But because this word is fetched by the preceding instruction, it is impossible to avoid this superfluous fetch.
5. In the case of an interrupt or trace exception, both words are not used.
6. The program counter usually points to the last word fetched from the instruction stream.

Instruction Execution Times

The following paragraphs contain listings of the instruction execution times in terms of external clock (CLK) periods. In this timing data, it is assumed that both memory read and write cycle times are four clock periods. Any wait states caused by a longer memory cycle must be added to the total instruction time. The number of bus read and write cycles for each instruction is also included with the timing data. This timing data is enclosed in parentheses following the execution periods and is shown as (r/w) where r is the number of read cycles and w is the number of write cycles.

NOTE:
The number of periods includes instruction fetch and all applicable operand fetches and stores.

Table 23. Effective Address Calculation Times

ADDRESSING MODE		BYTE,WORD	LONG
Register			
Dn	Data register direct	0(0/0)	0(0/0)
An	Address register direct	0(0/0)	0(0/0)
Memory			
(An)	Address register indirect	4(1/0)	8(2/0)
(An)+	Address register indirect with postincrement	4(1/0)	8(2/0)
−(An)	Address register indirect with predecrement	6(1/0)	10(2/0)
d(An)	Address register indirect with displacement	8(2/0)	12(3/0)
d(An,ix)*	Address register indirect with index	10(2/0)	14(3/0)
xxx.W	Absolute short	8(2/0)	12(3/0)
xxx.L	Absolute long	12(3/0)	16(4/0)
d(PC)	Program counter with displacement	8(2/0)	12(3/0)
d(PC,ix)*	Program counter with index	10(2/0)	14(3/0)
#xxx	Immediate	4(1/0)	8(2/0)

NOTE:
*The size of the index register (ix) does not affect execution time.

Effective Address Operand Calculation Timing

Table 23 lists the number of clock periods required to compute an instruction's effective address. It includes fetching of any extension words, the address computation, and fetching of the memory operand. The number of bus read and write cycles is shown in parentheses as (r/w). Note there are no write cycles involved in processing the effective address.

Move Instruction Execution Times

Tables 24 and 25 indicate the number of clock periods for the move instruction. This data includes instruction fetch, operand reads, and operand writes. The number of bus read and write cycles is shown in parentheses as (r/w).

Table 24. MOVE Byte Instruction Execution Times

SOURCE	DESTINATION								
	Dn	An	(An)	(An)+	−(An)	d(An)	d(An,ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
An	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
(An)	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
(An)+	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
−(An)	10(2/0)	10(2/0)	14(2/1)	14(2/1)	14(2/1)	18(3/1)	20(3/1)	18(3/1)	22(4/1)
d(An)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d(An,ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
xxx.W	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
xxx.L	16(4/0)	16(4/0)	20(4/1)	20(4/1)	20(4/1)	24(5/1)	26(5/1)	24(5/1)	28(6/1)
d(PC)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d(PC,ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
#xxx	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)

NOTE:
*The size of the index register (ix) does not affect execution time.

Table 25. MOVE LONG Instruction Execution Times

SOURCE	DESTINATION								
	Dn	An	(An)	(An)+	-(An)	d(An)	d(An,ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	12(1/2)	12(1/2)	12(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
An	4(1/0)	4(1/0)	12(1/2)	12(1/2)	12(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
(An)	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
(An)+	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
-(An)	14(3/0)	14(3/0)	22(3/2)	22(3/2)	22(3/2)	26(4/2)	28(4/2)	26(4/2)	30(5/2)
d(An)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
d(An,ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
xxx.W	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
xxx.L	20(5/0)	20(5/0)	28(5/2)	28(5/2)	28(5/2)	32(6/2)	34(6/2)	32(6/2)	36(7/2)
d(PC)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(5/2)
d(PC,ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
#xxx	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)

NOTE:
*The size of the index register (ix) does not affect execution time.

Standard Instruction Execution Times

The number of clock periods shown in Table 26 indicates the time required to perform the operations, store the results, and read the next instruction. The number of bus read and write cycles is shown in parentheses as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

In Table 26 the headings have the following meanings: An = address register operand, DN=data register operand, ea=an operand specified by an effective address, and M = memory effective address operand.

Table 26. Standard Instruction Execution Times

INSTRUCTION	SIZE	op<ea>, An†	op<ea>, Dn	op Dn, <M>
ADD	byte, word	8(1/0)+	4(1/0)+	8(1/1)+
	long	6(1/0)+ **	6(1/0)+ **	12(1/2)+
AND	byte, word	–	4(1/0)+	8(1/1)+
	long	–	6(1/0+ **	12(1/2)+
CMP	byte, word	6(1/0)+	4(1/0)+	–
	long	6(1/0)+	6(1/0)+	–
DIVS	–	–	158(1/0)+ *	–
DIVU	–	–	140(1/0)+ *	–
EOR	byte, word	–	4(1/0)***	8(1/1)+
	long	–	8(1/0)***	12(1/2)+
MULS	–	–	70(1/0)+ *	–
MULU	–	–	70(1/0)+ *	–
OR	byte, word	–	4(1/0)+	8(1/1)+
	long	–	6(1/0)+ **	12(1/2)+
SUB	byte, word	8(1/0)+	4(1/0)+	8(1/1)+
	long	6(1/0)+ **	6(1/0)+ **	12(1/2)+

NOTES:
+ Add effective address calculation time.
† Word or long word only.
* Indicates maximum value.
** The base time of 6 clock periods is increased to 8 if the effective address mode is register direct or immediate (effective address time should also be added).
*** Only available effective address mode is data register direct.
DIVS, DIVU – The divide algorithm used by the SCN68000 provides less than 10% difference between the best and worst case timings.
MULS, MULU – The multiply algorithm requires 38 + 2n clocks where n is defined as:
MULU: n = the number of ones in the < ea >.
MULS: n = concatenate the < ea > with a zero as the LSB; n is the resultant number of 10 or 01 patterns in the 17-bit source; i.e., worst case happens when the source is $5555.

Immediate Instruction Execution Times

The number of clock periods shown in Table 27 includes the time to fetch immediate operands, perform the operations, store the results, and read the next operation. The number of bus read and write cycles is shown in parentheses as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

In Table 27, the headings have the following meanings: # = immediate operand, DN = data register operand, An = address register operand, M = memory operand, and SR = status register.

Table 27. Immediate Instruction Execution Times

INSTRUCTION	SIZE	op#, Dn	op#, An	op#, M
ADDI	byte, word	8(2/0)	–	12(2/1)+
	long	16(3/0)	–	20(3/2)+
ADDQ	byte, word	4(1/0)	8(1/0)*	8(1/1)+
	long	8(1/0)	8(1/0)	12(1/2)+
ANDI	byte, word	8(2/0)	–	12(2/1)+
	long	16(3/0)	–	20(3/1)+
CMPI	byte, word	8(2/0)	–	8(2/0)+
	long	14(3/0)	–	12(3/0)+
EORI	byte, word	8(2/0)	–	12(2/1)+
	long	16(3/0)	–	20(3/2)+
MOVEQ	long	4(1/0)	–	–
ORI	byte, word	8(2/0)	–	12(2/1)+
	long	16(3/0)	–	20(3/2)+
SUBI	byte, word	8(2/0)	–	12(2/1)+
	long	16(3/0)	–	20(3/2)+
SUBQ	byte, word	4(1/0)	8(1/0)*	8(1/1)+
	long	8(1/0)	8(1/0)	12(1/2)+

NOTES:
+ Add effective address calculation time
* Word only

Single Operand Instruction Execution Times

Table 28 indicates the number of clock periods for the single operand instructions. The number of bus read and write cycles is shown in parentheses as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 28. Single Operand Instruction Execution Times

INSTRUCTION	SIZE	REGISTER	MEMORY
CLR	byte, word	4(1/0)	8(1/1)+
	long	6(1/0)	12(1/2)+
NBCD	byte	6(1/0)	8(1/1)+
NEG	byte, word	4(1/0)	8(1/1)+
	long	6(1/0)	12(1/2)+
NEGX	byte, word	4(1/0)	8(1/1)+
	long	6(1/0)	12(1/2)+
NOT	byte, word	4(1/0)	8(1/1)+
	long	6(1/0)	12(1/2)+
S_{CC}	byte, false	4(1/0)	8(1/1)+
	byte, true	6(1/0)	8(1/1)+
TAS	byte	4(1/0)	10(1/1)
TST	byte, word	4(1/0)	4(1/0)+
	long	4(1/0)	4(1/0)+

NOTE:
+ Add effective address calculation time

Shift/Rotate Instruction Execution Times

Table 29 indicates the number of clock periods for the shift and rotate instructions. The number of bus read and write cycles is shown in parentheses as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 29. Shift/Rotate Instruction Execution Times

INSTRUCTION	SIZE	REGISTER	MEMORY
ASR, ASL	byte, word	6 + 2n(1/0)	8(1/1)+
	long	8 + 2n(1/0)	–
LSR, LSL	byte, word	6 + 2n(1/0)	8(1/1)+
	long	8 + 2n(1/0)	–
ROR, ROL	byte, word	6 + 2n(1/0)	8(1/1)+
	long	8 + 2n(1/0)	–
ROXR, ROXL	byte, word	6 + 2n(1/0)	8(1/1)+
	long	8 + 2n(1/0)	–

NOTES:
+ Add effective address calculation time
n Is the shift or rotate count

Bit Manipulation Instruction Execution Times

Table 30 indicates the number of clock periods required for the bit manipulation instructions. The number of bus read and write cycles is shown in parentheses as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 30. Bit Manipulation Instruction Execution Times

INSTRUCTION	SIZE	DYNAMIC		STATIC	
		Register	Memory	Register	Memory
BCHG	byte	–	8(1/1)+	–	12(2/1)+
	long	8(1/0)*	–	12(2/0)*	–
BCLR	byte	–	8(1/1)+	–	12(2/1)+
	long	10(1/0)*	–	14(2/0)*	–
BSET	byte	–	8(1/1)+	–	12(2/1)+
	long	8(1/0)*	–	12(2/0)*	–
BTST	byte	–	4(1/0)+	–	8(2/0)+
	long	6(1/0)	–	10(2/0)	–

NOTES:
+ Add effective calculation time
* Indicates maximum value

Conditional Instruction Execution Times

Table 31 indicates the number of clock periods required for the conditional instructions. The number of bus read and write cycles is indicated in parentheses as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 31. Conditional Instruction Execution Times

INSTRUCTION	DISPLACEMENT	BRANCH TAKEN	BRANCH NOT TAKEN
B_{CC}	byte	10(2/0)	8(1/0)
	word	10(2/0)	12(2/0)
BRA	byte	10(2/0)	–
	word	10(2/0)	–
BSR	byte	18(2/2)	–
	word	18(2/2)	–
DB_{CC}	CC true	–	12(2/0)
	CC false	10(2/0)	14(3/0)

Table 32. JMP, JSR, LEA, PEA, and MOVEM Instruction Execution Times

INSTRUCTION	SIZE	(An)	(An)+	-(An)	d(An)	d(An, ix)+	xxx.W	xxx.L	d(PC)	d(PC,ix)*
JMP	–	8(2/0)	–	–	10(2/0)	14(3/0)	10(2/0)	12(3/0)	10(2/0)	14(3/0)
JSR	–	16(2/2)	–	–	18(2/2)	22(2/2)	18(2/2)	20(3/2)	18(2/2)	22(2/2)
LEA	–	4(1/0)	–	–	8(2/0)	12(2/0)	8(2/0)	12(3/0)	8(2/0)	12(2/0)
PEA	–	12(1/2)	–	–	16(2/2)	20(2/2)	16(2/2)	20(3/2)	16(2/2)	20(2/2)
MOVEM M → R	word	12 + 4n (3 + n/0)	12 + 4n (3 + n/0)	–	16 + 4n (4 + n/0)	18 + 4n (4 + n/0)	16 + 4n (4 + n/0)	20 + 4n (5 + n/0)	16 + 4n (4 + n/0)	18 + 4n (4 + n/0)
	long	12 + 8n (3 + 2n/0)	12 + 8n (3 + 2n/0)	–	16 + 8n (4 + 2n/0)	18 + 8n (4 + 2n/0)	16 + 8n (4 + 2n/0)	20 + 8n (5 + 2n/0)	16 + 8n (4 + 2n/0)	18 + 8n (4 + 2n/0)
MOVEM R → M	word	8 + 4n (2/n)	–	8 + 4n (2/n)	12 + 4n (3/n)	14 + 4n (3/n)	12 + 4n (3/n)	16 + 4n (4/n)	– –	– –
	long	8 + 8n (2/2n)	– –	8 + 8n (2/2n)	12 + 8n (3/2n)	14 + 8n (3/2n)	12 + 8n (3/2n)	16 + 8n (4/2n)	– –	– –

NOTES:
n Is the number of registers to move
* Is the size of the index register (ix) and does not affect the instruction's execution time

JMP, JSR, LEA, PEA, and MOVEM Instruction Execution Times

Table 32 indicates the number of clock periods required for the jump, jump-to-subroutine, load effective address, push effective address, and move multiple registers instructions. The number of bus read and write cycles is shown in parentheses as (r/w).

Multi-Precision Instruction Execution Times

Table 33 indicates the number of clock periods for the multi-precision instructions. The number of clock periods includes the time to fetch both operands, perform the operations, store the results, and read the next instructions. The number of read and write cycles is shown in parentheses as (r/w).

In Table 33, the headings have the following meanings: Dn = data register operand and M = memory operand.

Table 33. Multi-Precision Instruction Execution Times

INSTRUCTION	SIZE	op Dn, Dn	op M, M
ADDX	byte, word	4(1/0)	18(3/1)
	long	8(1/0)	30(5/2)
CMPM	byte, word	–	12(3/0)
	long	–	20(5/0)
SUBX	byte, word	4(1/0)	18(3/1)
	long	8(1/0)	30(5/2)
ABCD	byte	6(1/0)	18(3/1)
SBCD	byte	6(1/0)	18(3/1)

16-/32-Bit Microprocessor SCN68000

Table 34. Miscellaneous Instruction Execution Times

INSTRUCTION	SIZE	REGISTER	MEMORY
ANDI to CCR	byte	20(3/0)	–
ANDI to SR	word	20(3/0)	–
CHK	–	10(1/0)+	–
EORI to CCR	byte	20(3/0)	–
EORI to SR	word	20(3/0)	–
ORI to CCR	byte	20(3/0)	–
ORI to SR	word	20(3/0)	–
MOVE from SR	–	6(1/0)	8(1/1)+
MOVE to CCR	–	12(2/0)	12(2/0)+
MOVE to SR	–	12(2/0)	12(2/0)+
EXG	–	6(1/0)	–
EXT	word	4(1/0)	–
	long	4(1/0)	–
LINK	–	16(2/2)	–
MOVE from USP	–	4(1/0)	–
MOVE to USP	–	4(1/0)	–
NOP	–	4(1/0)	–
RESET	–	132(1/0)	–
RTE	–	20(5/0)	–
RTR	–	20(5/0)	–
RTS	–	16(4/0)	–
STOP	–	4(0/0)	–
SWAP	–	4(1/0)	–
TRAPV	–	4(1/0)	–
UNLK	–	12(3/0)	–

NOTE:
+ Add effective address calculation time

Table 35. MOVE PERIPHERAL Instruction Execution Times

INSTRUCTION	SIZE	REGISTER → MEMORY	MEMORY → REGISTER
MOVEP	word	16(2/2)	16(4/0)
	long	24(2/4)	24(6/0)

Table 36. Exception Processing Execution Times

EXCEPTION	PERIODS
Address error	50(4/7)
Bus error	50(4/7)
CHK instruction	44(5/4)+
Divide by zero	42(5/4)
Illegal instruction	34(4/3)
Interrupt	44(5/3)*
Privilege violation	34(4/3)
$\overline{\text{RESET}}$**	40(6/0)
Trace	34(4/3)
TRAP instruction	38(4/4)
TRAPV instruction	34(4/3)

NOTES:
+ Add effective address calculation time.
*The interrupt acknowledge cycle is assumed to take four clock periods.
**Indicates the time from when $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are first sampled as negated to when instruction execution starts.

Miscellaneous Instruction Execution Times

Tables 34 and 35 indicate the number of clock periods for the following miscellaneous instructions. The number of bus read and write cycles is shown in parentheses as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Exception Processing Execution Times

Table 36 indicates the number of clock periods for exception processing. The number of clock periods includes the time for all stacking, the vector fetch, and the fetch of the first two instruction words of the handler routine. The number of bus read and write cycles is shown in parentheses as (r/w).

ABSOLUTE MAXIMUM RATINGS

SYMBOL	RATING	VALUE	UNIT
V_{CC}	Supply voltage range	–0.3 to +7.0	V
V_{IN}	Input voltage range	–0.3 to +7.0	V
T_A	Operating temperature range SCN68000 SCN68000 ceramic	T_L to T_H 0 to 70 –40 to 85	°C
T_{TSG}	Storage temperature range	–55 to 150	°C

NOTE:
This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{CC}).

THERMAL CHARACTERISTICS

CHARACTERISTIC	SYMBOL	VALUE	SYMBOL	VALUE	RATING
Thermal Resistance (Still Air)					
Ceramic	θ_{JA}	30	θ_{JC}	15*	°C/W
Pin grid array		33		15	
Plastic		30		15*	

NOTE:
*Estimated

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V_{DC} \pm 5\%$; GND = $0V_{DC}$; $T_A = T_L$ to T_H (see Figures 42, 43, and 44)

SYMBOL	CHARACTERISTIC	SYMBOL	MIN	MAX
V_{IH}	Input high voltage	2	V_{CC}	V
V_{IL}	Input low voltage	GND − 0.3	0.8	V
I_{IN}	Input leakage current @ 5.25V $\overline{\text{BERR}}$, $\overline{\text{BGACK}}$, $\overline{\text{BR}}$, $\overline{\text{DTACK}}$, CLK, $\overline{\text{IPL0}}$-$\overline{\text{IPL2}}$, $\overline{\text{VPA}}$ $\overline{\text{HALT}}$, $\overline{\text{RESET}}$	2.5 –	μA 20	
I_{TSI}	3-State (off state) input current @ 2.4V/0.4V $\overline{\text{AS}}$, A1-A23, D0-D15, FC0-FC2, $\overline{\text{LDS}}$, R/$\overline{\text{W}}$, $\overline{\text{UDS}}$, $\overline{\text{VMA}}$	–	20	μA
V_{OH}	Output high voltage ($I_{OH} = -400\mu A$) E* E, $\overline{\text{AS}}$, A1-A23, $\overline{\text{BG}}$, D0-D15, FC0-FC2, $\overline{\text{LDS}}$, R/$\overline{\text{W}}$, $\overline{\text{UDS}}$, $\overline{\text{VMA}}$ 2.4	$V_{CC} - 0.75$ –	–	V
V_{OL}	Output low voltage ($I_{OL} = 1.6mA$) $\overline{\text{HALT}}$ ($I_{OL} = 3.2mA$) A1-A23, $\overline{\text{BG}}$, FC0-FC2 ($I_{OL} = 5.0mA$) $\overline{\text{RESET}}$ ($I_{OL} = 5.3mA$) E, $\overline{\text{AS}}$, D0-D15, $\overline{\text{LDS}}$, R/$\overline{\text{W}}$, $\overline{\text{UDS}}$, $\overline{\text{VMA}}$	 – – – –	 0.5 0.5 0.5 0.5	V
P_D***	Power dissipation (See Power Considerations)	–	–	W
C_{IN}	Capacitance ($V_{IN} = 0V$, $T_A = 25°C$; frequency = 1MHz**)	–	20	pF

NOTES:

* With external pullup resistor of 1.1kΩ.

** Capacitance is periodically sampled rather than 100% tested.

*** During normal operation instantaneous V_{CC} current requirements may be as high as 1.5A.

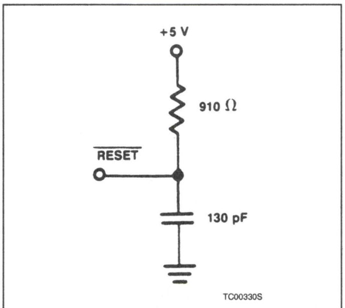

+5 V
910 Ω
$\overline{\text{RESET}}$
130 pF
TC00330S

Figure 42. $\overline{\text{RESET}}$ Test Load

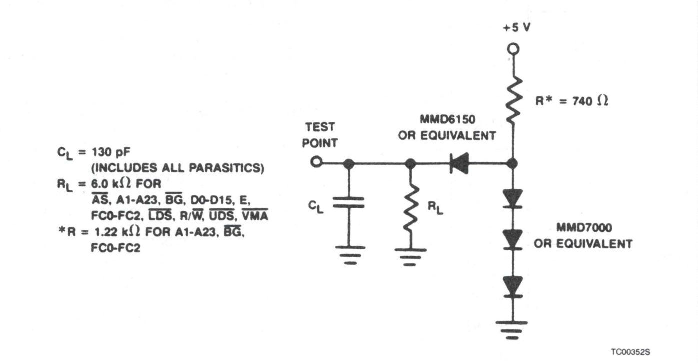

+5 V
R* = 740 Ω
TEST POINT
MMD6150 OR EQUIVALENT
C_L = 130 pF (INCLUDES ALL PARASITICS)
R_L = 6.0 kΩ FOR $\overline{\text{AS}}$, A1-A23, $\overline{\text{BG}}$, D0-D15, E, FC0-FC2, $\overline{\text{LDS}}$, R/$\overline{\text{W}}$, $\overline{\text{UDS}}$, $\overline{\text{VMA}}$
*R = 1.22 kΩ FOR A1-A23, $\overline{\text{BG}}$, FC0-FC2
C_L
R_L
MMD7000 OR EQUIVALENT
TC00352S

Figure 44. Test Loads

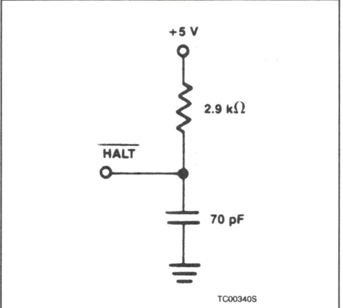

+5 V
2.9 kΩ
$\overline{\text{HALT}}$
70 pF
TC00340S

Figure 43. $\overline{\text{HALT}}$ Test Load

Power Considerations

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = ambient temperature, °C

θ_{JA} = package thermal resistance, junction-to-ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{CC} \times V_{CC}$, watts – chip internal power

$P_{I/O}$ = power dissipation on input and output pins — user determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = T_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

Figure 45 illustrates the graphic solution to the equations, given above, for the specification power dissipations of 1.50 and 1.75 watts over the ambient temperature range of −55°C to 125°C using an average θ_{JA} of 40°C/watt to represent various SCN68000 packages.

However, actual θ_{JA}'s in the range of 30°C to 50°C/watt only change the curves slightly.

The total thermal resistance of a package (θ_{JA}) can be separated into two components, θ_{JC} and θ_{CA}, representing the barrier to heat flow from the semiconductor junction to the package (case) surface (θ_{JC}) and from the case to the outside ambient (θ_{CA}). These terms are related by the equation:

$$\theta_{JA} = \theta_{JC} + \theta_{CA} \qquad (4)$$

θ_{JC} is device related and cannot be influenced by the user. However, θ_{CA} is user dependent and can be minimized by such thermal management techniques as heat sinks, ambient air cooling and thermal convention. Thus, good thermal management on the part of the user can significantly reduce θ_{CA} so that θ_{JA} approximately equals θ_{JC}. Substitution of θ_{JC} for θ_{JA} in equation (1) will result in a lower semiconductor junction temperature.

Values for thermal resistance presented in this data sheet are provided for design purposes only. Thermal measurements are complex and dependent on procedure and setup. User derived values for thermal resistance may differ.

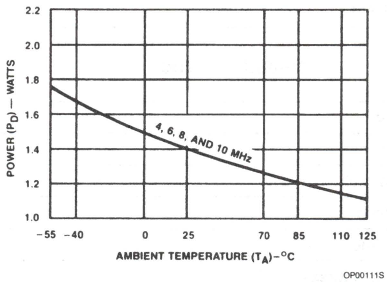


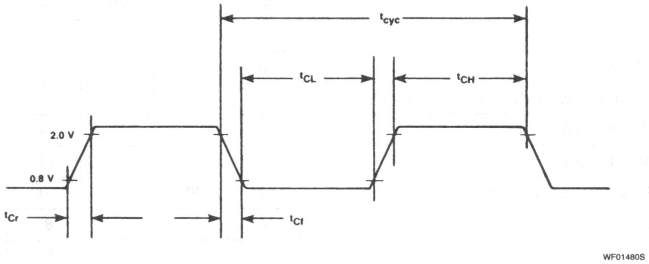
Figure 45. SCN68000 Power Dissipation (P_D) vs Ambient Temperature (T_A)

AC ELECTRICAL CHARACTERISTICS — Clock Timing (see Figure 46)

SYMBOL	CHARACTERISTIC	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
f	Frequency of operation	4	8	4	10	4	12.5	MHz
t_{cyc}	Cycle time	125	250	100	250	80	250	ns
t_{CL} t_{CH}	Clock pulse width	55 55	125 125	45 45	125 125	35 35	125 125	ns
t_{Cr} t_{Cf}	Rise and fall times	– –	10 10	– –	10 10	– –	5 5	ns

Table 37. Maximum Power Dissipation by Package Type Modes

PACKAGE TYPE	TEMPERATURE (°C)	MAXIMUM POWER DISSIPATION (WATTS) PER FREQUENCY (MHz)		
		8MHz	10MHz	12.5MHz
Ceramic	0 to 70 −40 to 85	1.50 1.65	1.50 1.65	1.75 –
Plastic	0 to 70 −40 to 85	1.50 1.65	1.50 1.65	– –
Pin grid array	0 to 70 −40 to 85	1.50 1.65	1.50 1.65	1.75 –
Plastic LCC	0 to 70	1.50	1.50	–



NOTE:
Timing measurements are referenced to and from a low voltage of 0.8V and a high voltage of 2V, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8V and 2V.

Figure 46. Clock Input Timing Diagram

AC ELECTRICAL CHARACTERISTICS — Read and Write Cycles $V_{CC} = 5V_{DC} \pm 5\%$; GND = $0V_{DC}$; $T_A = T_L$ to T_H (see Figures 47 and 48)

NO.	CHARACTERISTIC	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
1	Clock period	125	250	100	250	80	250	ns
2	Clock width low	55	125	45	125	35	125	ns
3	Clock width high	55	125	45	125	35	125	ns
4	Clock fall time	–	10	–	10	–	5	ns
5	Clock rise time	–	10	–	10	–	5	ns
6	Clock low to address valid	–	70	–	60	–	55	ns
6A	Clock high to FC valid	–	70	–	60	–	55	ns
7	Clock high to address data high impedance (maximum)	–	80	–	70	–	60	ns
8	Clock high to address/FC invalid (minimum)	0	–	0	–	0	–	ns
9[1]	Clock high to $\overline{\text{AS}}$, $\overline{\text{DS}}$ low	0	60	0	55	0	55	ns
11[2]	Address valid to $\overline{\text{AS}}$, $\overline{\text{DS}}$ low (read)/$\overline{\text{AS}}$ low write	30	–	20	–	0	–	ns
11A[2,7]	FC valid to $\overline{\text{AS}}$, $\overline{\text{DS}}$ low (read)/$\overline{\text{AS}}$ low (write)	60	–	50	–	40	–	ns
12[1]	Clock low to $\overline{\text{AS}}$, $\overline{\text{DS}}$ high	–	70	–	55	–	50	ns
13[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to address/FC invalid	30	–	20	–	10	–	ns
14[2,5]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ width low (read)/$\overline{\text{AS}}$ low (write)	240	–	195	–	160	–	ns
14A[2]	$\overline{\text{DS}}$ width low (write)	115	–	95	–	80	–	ns
15[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ width high	150	–	105	–	65	–	ns
16	Clock high to control bus high impedance	–	80	–	70	–	60	ns
17[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to R/$\overline{\text{W}}$ high (read)	40	–	20	–	10	–	ns
18[1]	Clock high to R/$\overline{\text{W}}$ high	0	70	0	60	0	60	ns
20[1]	Clock high to R/$\overline{\text{W}}$ low (write)	–	70	–	60	–	60	ns
20A[8]	$\overline{\text{AS}}$ low to R/$\overline{\text{W}}$ valid	–	20	–	20	–	20	ns
21[2]	Address valid to R/$\overline{\text{W}}$ low (write)	20	–	0	–	0	–	ns
21A[2,7]	FC valid to R/$\overline{\text{W}}$ low (write)	60	–	50	–	30	–	ns
22[2]	R/$\overline{\text{W}}$ low to $\overline{\text{DS}}$ low (write)	80	–	50	–	30	–	ns
23	Clock low to data out valid (write)	–	70	–	55	–	55	ns
25[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to data out invalid (write)	30	–	20	–	15	–	ns
26[2]	Data out valid to $\overline{\text{DS}}$ low (write)	30	–	20	–	15	–	ns
27[6]	Data in to clock low (setup time on read)	15	–	10	–	10	–	ns
28[2,5]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to $\overline{\text{DTACK}}$ high	0	245	0	190	0	150	ns
29	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to data in invalid (hold time on read)	0	–	0	–	0	–	ns
30	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to $\overline{\text{BERR}}$ high	0	–	0	–	0	–	ns
31[2,6]	$\overline{\text{DTACK}}$ low to data in (setup time)	–	90	–	65	–	50	ns
32	$\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ input transition time	0	200	0	200	0	200	ns
33	Clock high to $\overline{\text{BG}}$ low	–	70	–	60	–	50	ns
34	Clock high to $\overline{\text{BG}}$ high	–	70	–	60	–	50	ns
35	$\overline{\text{BR}}$ low to $\overline{\text{BG}}$ low	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	Clk. per.
36[9]	$\overline{\text{BR}}$ high to $\overline{\text{BG}}$ high	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	Clk. per.

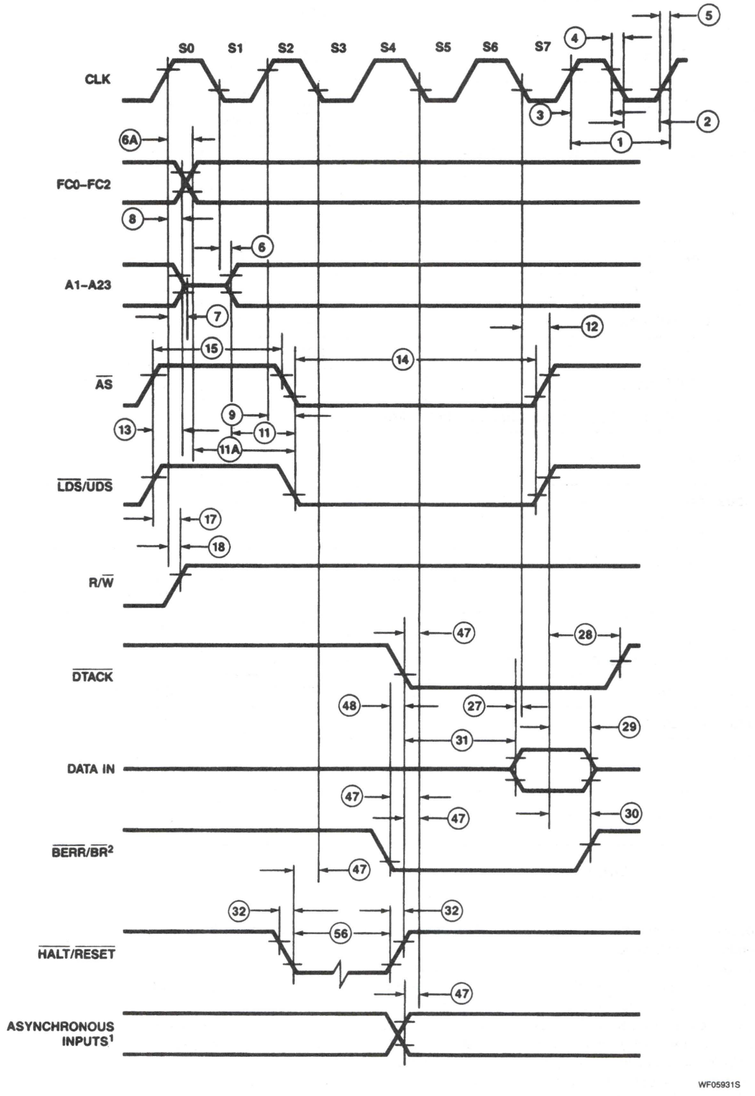
AC ELECTRICAL CHARACTERISTICS — Read and Write (Continued)

NO.	CHARACTERISTIC	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
37	$\overline{\text{BGACK}}$ low to $\overline{\text{BG}}$ low	1.5	90ns +3	1.5	80ns +3	1.5	70ns +3	Clk. per.
37A[10]	$\overline{\text{BGACK}}$ low to $\overline{\text{BR}}$ high	20	1.5 clocks	20	1.5 clocks	20	1.5 clocks	ns
38	$\overline{\text{BG}}$ low to control, address, data bus high impedance ($\overline{\text{AS}}$ high)	100	–	80	–	70	–	60
39	$\overline{\text{BG}}$ width high	1.5	–	1.5	–	1.5	–	Clk. per.
40	Clock low to $\overline{\text{VMA}}$ low	–	70	–	70	–	70	ns
41	Clock low to E transition	–	70	–	55	–	45	ns
42	E output rise and fall time	–	25	–	25	–	25	ns
43	$\overline{\text{VMA}}$ low to E high	200	–	150	–	90	–	ns
44	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to $\overline{\text{VPA}}$ high	0	120	0	90	0	70	ns
45	E low to control, address bus invalid (address hold time)	30	–	10	–	10	–	ns
46	$\overline{\text{BGACK}}$ width low	1.5	–	1.5	–	1.5	–	Clk. per.
47[6]	Asynchronous input setup time	20	–	20	–	20	–	ns
48[3]	$\overline{\text{BERR}}$ low to $\overline{\text{DTACK}}$ low	20	–	20	–	20	–	ns
49[11]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to E low	–70	70	–55	55	–45	45	ns
50	E width high	450	–	350	–	280	–	ns
51	E width low	700	–	550	–	440	–	ns
53	Clock high to data out invalid	0	–	0	–	0	–	ns
54	E low to data out invalid	30	–	20	–	15	–	ns
55	R/$\overline{\text{W}}$ to data bus driver	30	–	20	–	10	–	ns
56[4]	$\overline{\text{HALT}}$/$\overline{\text{RESET}}$ pulse width	10	–	10	–	10	–	Clk. per.
57	$\overline{\text{BGACK}}$ high to control bus driven	1.5	–	1.5	–	1.5	–	Clk. per.
58[9]	$\overline{\text{BG}}$ high to control bus driven	1.5	–	1.5	–	1.5	–	Clk. per.

NOTES:

1. For a loading capacitance of less than or equal to 50pF, subtract 5ns from the values given in these columns.
2. Actual value depends on clock period.
3. If #47 is satisfied for both $\overline{\text{DTACK}}$ and $\overline{\text{BERR}}$, #48 may be 0ns.
4. For power up, the MPU must be held in $\overline{\text{RESET}}$ state for 100ms to allow stabilization of on-chip circuitry. After the system is powered up, #56 refers to the minimum pulse width required to reset the system.
5. #14, #14A, and #28 are one clock period less than the given number for T6E, BF4, and R9M mask sets.
6. If the asynchronous setup time (#47) requirements are satisfied, the $\overline{\text{DTACK}}$ low-to-data setup time (#31) requirement can be ignored. The data must only satisfy the data-in clock-low setup time (#27) for the following cycle.
7. For T6E, BF4, and R9M mask set #11A timing equals #11, and #21A equals #21. #20A may be 0 for T6E, BF4, and R9M mask sets.
8. When $\overline{\text{AS}}$ and R/$\overline{\text{W}}$ are equally loaded (±20%), subtract 10ns from the values given in these columns.
9. The processor will negate $\overline{\text{BG}}$ and begin driving the bus again if external arbitration logic negates $\overline{\text{BR}}$ before asserting $\overline{\text{BGACK}}$.
10. The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{\text{BG}}$ may be reasserted.
11. The falling edge of S6 triggers both the negation of the strobes ($\overline{\text{AS}}$ and $\overline{\text{xDS}}$) and the falling edge of E. Either of these events can occur first, depending upon the loading on each signal. Specification #49 indicates the absolute maximum skew that will occur between the rising edge of the strobes and the falling edge of the E clock.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

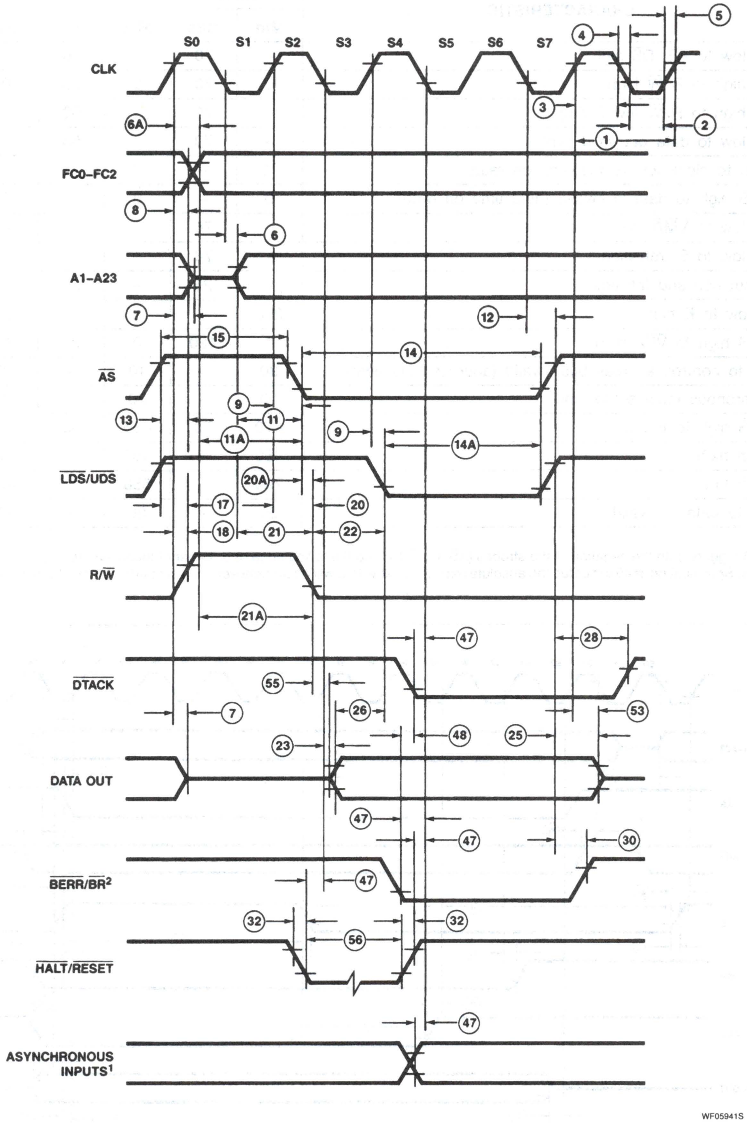

S0
S1
S2
S3
S4
S5
S6
S7
CLK
FC0–FC2
A1–A23
AS
LDS/UDS
R/W
DTACK
DATA IN
BERR/BR2
HALT/RESET
ASYNCHRONOUS INPUTS1
①
②
③
④
⑤
6A
⑥
⑦
⑧
⑨
⑪
11A
⑫
⑬
⑭
⑮
⑰
⑱
㉗
㉘
㉙
㉚
㉛
㉜
㊼
㊽
(56)
WF05931S

NOTES:

1. Setup time for the asynchronous inputs $\overline{BGACK}$, $\overline{IPL0\text{-}2}$, and $\overline{VPA}$ guarantees their recognition at the next falling edge of the clock.
2. $\overline{BR}$ need fall at this time only in order to insure being recognized at the end of this bus cycle.
3. Timing measurements are referenced to and from a low voltage of 0.8V and a high voltage of 2V, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8V and 2V.

Figure 47. Read Cycle Timing Diagram

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.


S0
S1
S2
S3
S4
S5
S6
S7
CLK
FC0–FC2
A1–A23
$\overline{AS}$
$\overline{LDS}/\overline{UDS}$
R/$\overline{W}$
$\overline{DTACK}$
DATA OUT
$\overline{BERR}/\overline{BR}^2$
$\overline{HALT}/\overline{RESET}$
ASYNCHRONOUS INPUTS¹
①
②
③
④
⑤
6A
⑥
⑦
⑧
⑨
11A
⑪
⑫
⑬
⑭
14A
⑮
⑰
⑱
20A
⑳
㉑
21A
㉒
㉓
㉕
㉖
㉘
㉚
㉜
㊼
㊽
(53)
(55)
(56)
WF05941S

NOTES:

1. Timing measurements are referenced to and from a low voltage of 0.8V and a high voltage of 2V, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8V and 2V.
2. Because of loading variations, R/$\overline{W}$ may be valid after $\overline{AS}$ even though both are initiated by the rising edge of S2 (Specification 20A).

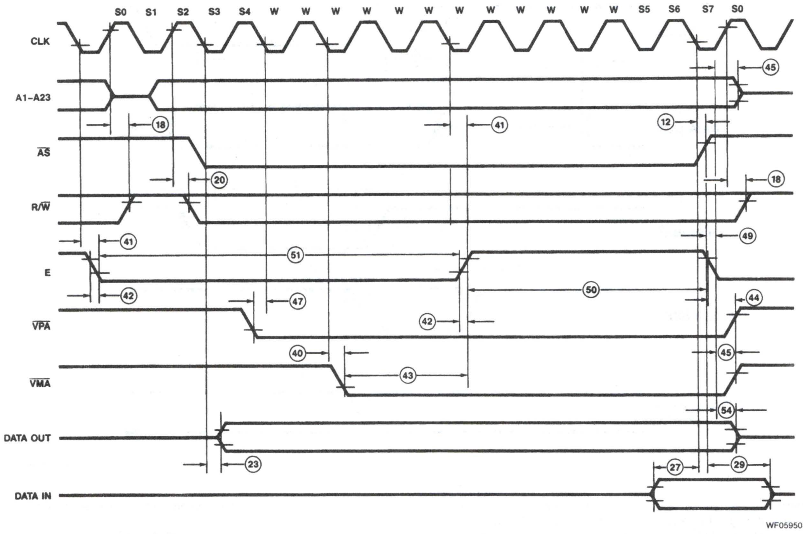
Figure 48. Write Cycle Timing Diagram

AC ELECTRICAL CHARACTERISTICS - SCN68000 To Synchronous Peripheral $V_{CC} = 5V_{DC} \pm 5\%$; GND = $0V_{DC}$, $T_A = T_L$ to T_H (refer to Figures 49 and 50)

NO.	CHARACTERISTIC	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
12	Clock low to $\overline{AS}$, $\overline{DS}$ high	–	70	–	55	–	50	ns
18	Clock high to R/$\overline{W}$ high	0	70	0	60	0	60	ns
20	Clock high to R/$\overline{W}$ low (write)	–	70	–	60	–	60	ns
23	Clock low to data out valid (write)	–	70	–	55	–	55	ns
27	Data in to clock low (setup time on read)	15	–	10	–	10	–	ns
29	$\overline{AS}$, $\overline{DS}$ high to data in invalid (hold time on read)	0	–	0	–	0	–	ns
40	Clock low to $\overline{VMA}$ low	–	70	–	70	–	70	ns
41	Clock low to E transition	–	70	–	55	–	45	ns
42	E output rise and fall time	–	25	–	25	–	25	ns
43	$\overline{VMA}$ low to E high	200	–	150	–	90	–	ns
44	$\overline{AS}$, $\overline{DS}$ high to $\overline{VPA}$ high	0	120	0	90	0	70	ns
45	E low to control, address bus invalid (address hold time)	30	–	10	–	10	–	ns
47	Asynchronous input setup time	20	–	20	–	20	–	ns
49[1]	$\overline{AS}$, $\overline{DS}$ high to E low	–70	70	–55	55	–45	45	ns
50	E width high	450	–	350	–	280	–	ns
51	E width low	700	–	550	–	440	–	ns
54	E low to data out valid	30	–	20	–	15	–	ns

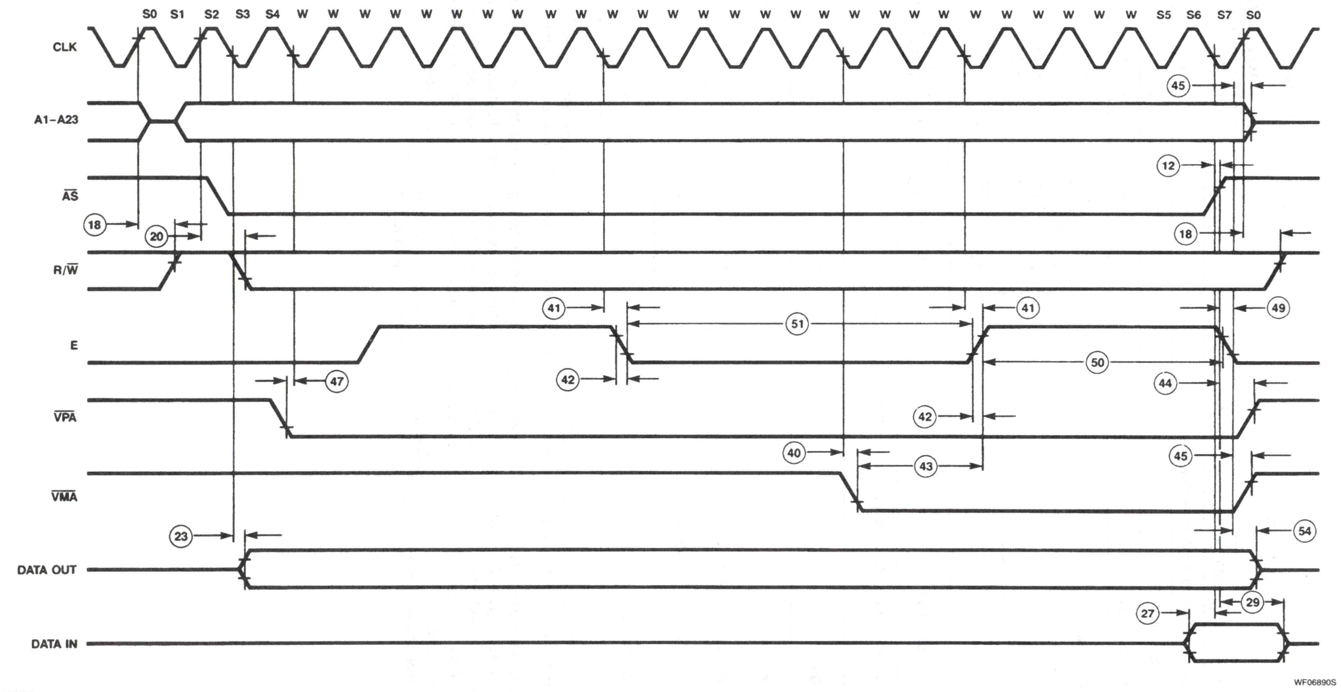
NOTE:

1. The falling edge of S6 triggers both the negation of the strobes ($\overline{AS}$ and $\overline{xDS}$) and the falling edge of E. Either of these events can occur first, depending upon the loading on each signal. Specification #49 indicates the absolute maximum skew that will occur between the rising edge of the strobes and the falling edge of the E clock.



NOTE:
This timing diagram is included for those who wish to design their own circuit to generate VMA. It shows the best case possibly attainable.

Figure 49. SCN68000 to Synchronous Peripheral Timing Diagram—Best Case


S0 S1 S2 S3 S4 W W W W W W W W W W W W W W W W W W W W W W W W W W S5 S6 S7 S0
CLK
A1–A23
AS
R/W
E
VPA
VMA
DATA OUT
DATA IN
⑱
⑳
㉓
㊼
㊶
㊷
(51)
㊵
㊸
(50)
㊹
㊺
⑫
(49)
(54)
㉗
㉙
WF06890S

NOTE:
This timing diagram is included for those who wish to design their own circuit to generate VMA. It shows the worst case possibly attainable.

Figure 50. SCN68000 to Synchronous Peripheral Timing Diagram—Worst Case

AC ELECTRICAL CHARACTERISTICS — Bus Arbitration $V_{CC} = 5V_{DC} \pm 5\%$; $GND = 0V_{DC}$; $T_A = T_L$ to T_H (see Figures 51, 52, 53)

NO.	CHARACTERISTIC	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
7	Clock high to address, data bus high impedence	–	80	–	70	–	60	ns
16	Clock high to control bus high impedence	–	80	–	70	–	60	ns
33	Clock high to $\overline{\text{BG}}$ low	–	70	–	60	–	50	ns
34	Clock high to $\overline{\text{BG}}$ high	–	70	–	60	–	50	ns
35	$\overline{\text{BR}}$ low to $\overline{\text{BG}}$ low	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	Clk. per.
36[1]	$\overline{\text{BR}}$ high to $\overline{\text{BG}}$ high	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	Clk. per.
37	$\overline{\text{BGACK}}$ low to $\overline{\text{BG}}$ high	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	Clk. per.
37A[2]	$\overline{\text{BGACK}}$ low to $\overline{\text{BR}}$ high	20	1.5 Clocks	20	1.5 Clocks	20	1.5 Clocks	ns
38	$\overline{\text{BG}}$ low to control, address, data bus high impedance ($\overline{\text{AS}}$ high)	–	80	–	70	–	60	ns
39	$\overline{\text{BG}}$ width high	1.5	–	1.5	–	1.5	–	Clk. per.
46	$\overline{\text{BGACK}}$ width low	1.5	–	1.5	–	1.5	–	Clk. per.
47	Asynchronous input setup time	20	–	20	–	20	–	ns
57	$\overline{\text{BGACK}}$ high to control bus driven	1.5	–	1.5	–	1.5	–	Clk. per.
58[1]	$\overline{\text{BG}}$ high to control bus driven	1.5	–	1.5	–	1.5	–	Clk. per.

NOTES:

1. The processor will negate $\overline{\text{BG}}$ and begin driving the bus again if external arbitration logic negates $\overline{\text{BR}}$ before asserting $\overline{\text{BGACK}}$.
2. The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{\text{BG}}$ may be reasserted.

Figures 51, 52, and 53 depict the three bus arbitration cases that can arise. Figure 51 shows the timing where $\overline{\text{AS}}$ is negated when the processor asserts $\overline{\text{BG}}$ (idle bus case). Figure 52 shows the timing where $\overline{\text{AS}}$ is asserted when the processor asserts $\overline{\text{BG}}$ (active bus case). Figure 53 shows the timing where more than one bus master are requesting the bus. Refer to **Bus Arbitration** for a complete discussion of bus arbitration.

The waveforms shown in Figures 51, 52, and 53 should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

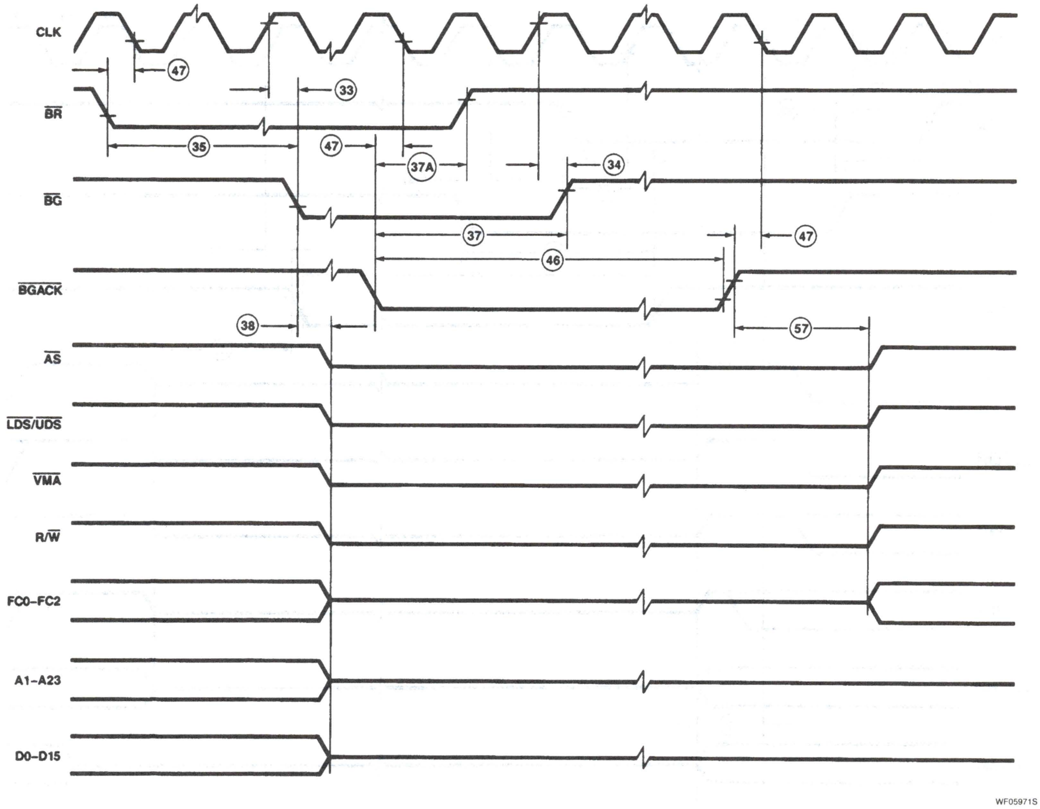

CLK
BR
BG
BGACK
AS
LDS/UDS
VMA
R/W
FC0–FC2
A1–A23
D0–D15
47
33
35
47
37A
34
37
47
46
38
57
WF05971S

Figure 51. Bus Arbitration Timing Diagram — Idle Bus Case

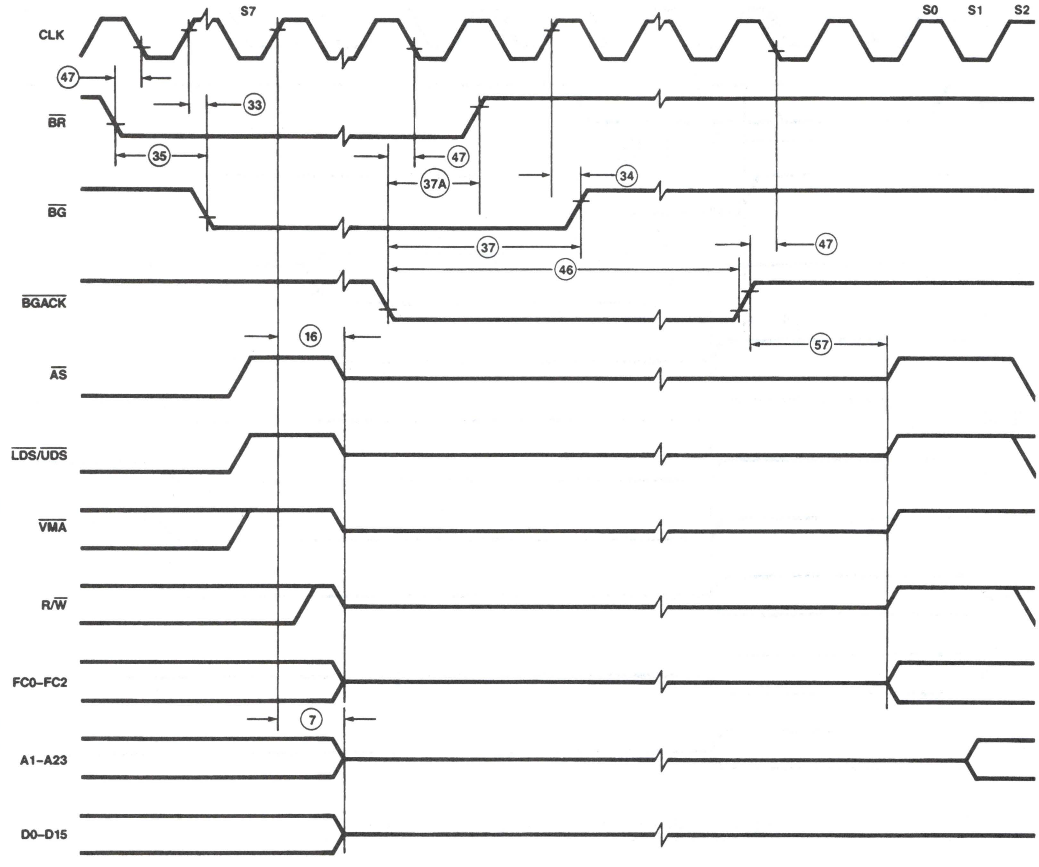

CLK
S7
S0
S1
S2
BR
BG
BGACK
AS
LDS/UDS
VMA
R/W
FC0–FC2
A1–A23
D0–D15
47
33
35
47
37A
34
37
46
47
16
57
7
WF05980S

Figure 52. Bus Arbitration Timing Diagram — Active Bus Case

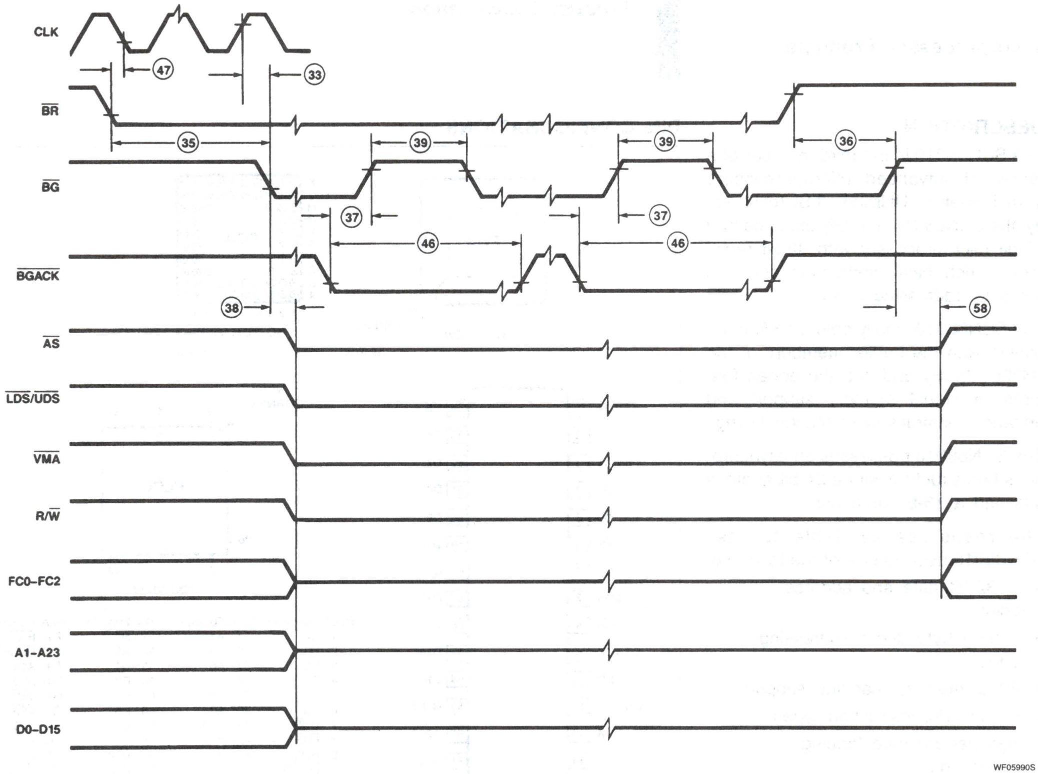

CLK
BR
BG
BGACK
AS
LDS/UDS
VMA
R/W
FC0–FC2
A1–A23
D0–D15
47
33
35
39
39
36
37
37
46
46
38
58
WF05990S

Figure 53. Bus Arbitration Timing Diagram — Multiple Bus Requests

Signetics

SCN68010
16-Bit Virtual Memory Microprocessor

Product Specification

Microprocessor Products

DESCRIPTION

The SCN68010 is the third member of a family of advanced microprocessors from Signetics. Utilizing VLSI technology, the SCN68010 is a fully-implemented 16-bit microprocessor with 32-bit registers, a rich basic instruction set, and versatile addressing modes.

The SCN68010 is fully object code compatible with the earlier members of the S68000 family and has the added features of virtual memory support and enhanced instruction execution timing.

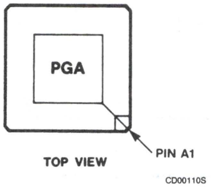
The SCN68010 possesses an asynchronous bus structure with a 24-bit address bus and a 16-bit data bus.

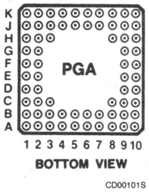
The resources available to the SCN68010 user consist of the following:

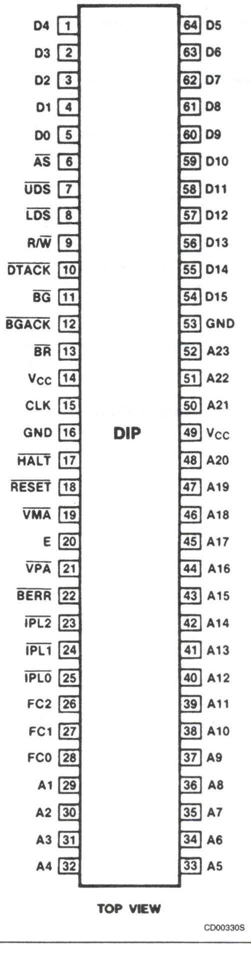
- 17 32-bit data and address registers
- 16-megabyte direct addressing range
- Virtual memory/machine support
- 57 powerful instruction types
- High performance looping instructions
- Operations on five main data types
- Memory mapped I/O
- 14 addressing modes

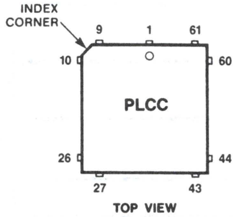
As shown in the programming model (Figures 1 and 2), the SCN68010 offers 17 32-bit general purpose registers, a 32-bit program counter, a 16-bit status register, a 32-bit vector base register, and two 3-bit alternate function code registers. The first eight registers (D0 – D7) are used as data registers for byte (8-bit), word (16-bit), and long word (32-bit) operations. The second set of seven registers (A0 – A6) and the stack pointers (SSP, USP) may be used as software stack pointers and base address registers. In addition, the address registers may be used for word and long word operations. All of the 17 registers may be used as index registers.

PIN CONFIGURATIONS


PGA
PIN A1
TOP VIEW
CD00110S


K
J
H
G
F
E
D
C
B
A
PGA
1 2 3 4 5 6 7 8 9 10
BOTTOM VIEW
CD00101S


D4 1
D3 2
D2 3
D1 4
D0 5
$\overline{AS}$ 6
$\overline{UDS}$ 7
$\overline{LDS}$ 8
R/$\overline{W}$ 9
$\overline{DTACK}$ 10
$\overline{BG}$ 11
$\overline{BGACK}$ 12
$\overline{BR}$ 13
V_{CC} 14
CLK 15
GND 16
$\overline{HALT}$ 17
$\overline{RESET}$ 18
$\overline{VMA}$ 19
E 20
$\overline{VPA}$ 21
$\overline{BERR}$ 22
$\overline{IPL2}$ 23
$\overline{IPL1}$ 24
$\overline{IPL0}$ 25
FC2 26
FC1 27
FC0 28
A1 29
A2 30
A3 31
A4 32
64 D5
63 D6
62 D7
61 D8
60 D9
59 D10
58 D11
57 D12
56 D13
55 D14
54 D15
53 GND
52 A23
51 A22
50 A21
49 V_{CC}
48 A20
47 A19
46 A18
45 A17
44 A16
43 A15
42 A14
41 A13
40 A12
39 A11
38 A10
37 A9
36 A8
35 A7
34 A6
33 A5
DIP
TOP VIEW
CD00330S


INDEX CORNER
9
1
61
10
60
PLCC
26
44
27
43
TOP VIEW
CD0068PS

PLCC	Function	PLCC	Function	PGA	Function	PGA	Function
1	D4	35	A4	A1	NC	F1	$\overline{HALT}$
2	D3	36	A5	A2	AS	F2	$\overline{RESET}$
3	D2	37	A6	A3	D1	F9	A18
4	D1	38	A7	A4	D2	F10	A19
5	D0	39	A8	A5	D4	G1	$\overline{VMA}$
6	AS	40	A9	A6	D5	G2	$\overline{VPA}$
7	$\overline{UDS}$	41	A10	A7	D7	G9	A15
8	$\overline{LDS}$	42	A11	A8	D8	G10	A17
9	R/$\overline{W}$	43	A12	A9	D10	H1	E
10	$\overline{DTACK}$	44	A13	A10	D12	H2	$\overline{IPL2}$
11	$\overline{BG}$	45	A14	B1	$\overline{DTACK}$	H3	$\overline{IPL1}$
12	$\overline{BGACK}$	46	A15	B2	$\overline{LDS}$	H8	A13
13	$\overline{BR}$	47	A16	B3	$\overline{UDS}$	H9	A12
14	V_{CC}	48	A17	B4	D0	H10	A16
15	CLK	49	A18	B5	D3	J1	$\overline{BERR}$
16	GND	50	A19	B6	D6	J2	$\overline{IPL0}$
17	GND	51	A20	B7	D9	J3	FC1
18	NC	52	V_{CC}	B8	D11	J4	NC
19	$\overline{HALT}$	53	A21	B9	D14	J5	A2
20	$\overline{RESET}$	54	A22	B10	D15	J6	A5
21	$\overline{VMA}$	55	A23	C1	$\overline{BGACK}$	J7	A8
22	E	56	GND	C2	$\overline{BG}$	J8	A10
23	$\overline{VPA}$	57	GND	C3	R/$\overline{W}$	J9	A11
24	$\overline{BERR}$	58	D15	C8	D13	J10	A14
25	$\overline{IPL2}$	59	D14	C9	A23	K1	NC
26	$\overline{IPL1}$	60	D13	C10	A22	K2	FC2
27	$\overline{IPL0}$	61	D12	D1	$\overline{BR}$	K3	FC0
28	FC2	62	D11	D2	V_{CC}	K4	A1
29	FC1	63	D10	D9	V_{SS}	K5	A3
30	FC0	64	D9	D10	A21	K6	A4
31	NC	65	D8	E1	CLK	K7	A6
32	A1	66	D7	E2	V_{SS}	K8	A7
33	A2	67	D6	E9	V_{CC}	K9	A9
34	A3	68	D5	E10	A20	K10	NC

16-Bit Virtual Memory Microprocessor SCN68010

ORDERING INFORMATION

PACKAGES	$V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to 70°C		
	8MHz	10MHz	12.5MHz
Ceramic DIP	SCN68010C8I64	SCN68010CAI64	SCN68010CBI64
Plastic DIP	SCN68010C8N64	SCN68010CAN64	SCN68010CBN64
Plastic LCC	SCN68010C8A68	SCN68010CAA68	SCN68010CBA68
Pin Grid Array	SCN68010C8P68	SCN68010CAP68	SCN68010CBP68

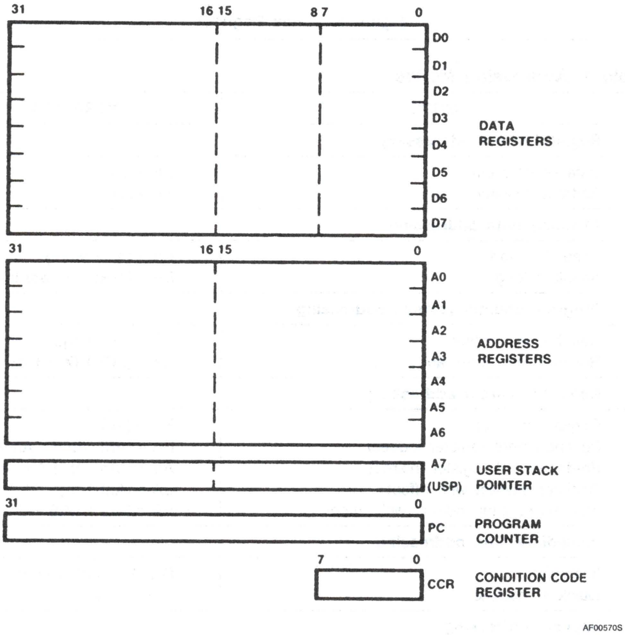

31
16 15
8 7
0
D0
D1
D2
D3
D4
D5
D6
D7
DATA REGISTERS
A0
A1
A2
A3
A4
A5
A6
ADDRESS REGISTERS
A7 (USP)
USER STACK POINTER
PC
PROGRAM COUNTER
7
CCR
CONDITION CODE REGISTER
AF00570S

Figure 1. User Programming Model

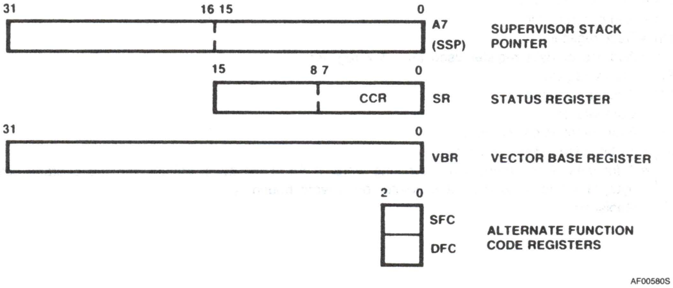

31
16 15
0
A7 (SSP)
SUPERVISOR STACK POINTER
15
8 7
CCR
SR
STATUS REGISTER
VBR
VECTOR BASE REGISTER
2
SFC
DFC
ALTERNATE FUNCTION CODE REGISTERS
AF00580S

Figure 2. Supervisor Programming Model Supplement

16-Bit Virtual Memory Microprocessor SCN68010

The status register (Figure 3) contains the interrupt mask (eight levels available) as well as the condition codes; extend (X), negative (N), zero (Z), overflow (V), and carry (C). Additional status bits indicate that the processor is in the trace (T) mode and in the supervisor (S) or user state.

The vector base register is used to determine the location of the exception vector table in memory to support multiple vector tables. The alternate function code registers allow the supervisor to access user data space or emulate CPU space cycles.

Data Types and Addressing Modes

Five basic data types are supported. These data types are:

- Bits
- BCD digits (4 bits)
- Bytes (8 bits)
- Words (16 bits)
- Long words (32 bits)

In addition, operations on other data types such as memory addresses, status word data, etc., are provided in the instruction set.

The 14 address modes, shown in Table 1, include six basic types:

- Register direct
- Register indirect
- Absolute
- Program counter relative
- Immediate
- Implied

Included in the register indirect addressing modes is the capability to do postincrementing, predecrementing, offsetting, and indexing. The program counter relative mode can also be modified via indexing and offsetting.

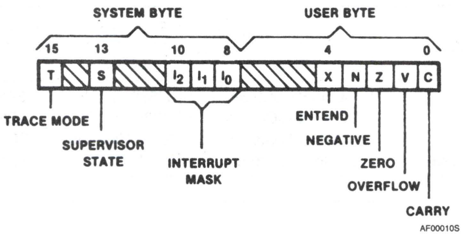


Figure 3. Status Register

Table 1. Addressing Modes

MODE	GENERATION
Register direct addressing	
Data register direct	EA = Dn
Address register direct	EA = An
Absolute data addressing	
Absolute short	EA = (next word)
Absolute long	EA = (next two words)
Program counter relative addressing	
Relative with offset	EA = (PC) + d_{16}
Relative with index and offset	EA = (PC) + (Xn) + d_8
Register indirect addressing	
Register indirect	EA = (An)
Postincrement register indirect	EA = (An), An ← An + N
Predecrement register indirect	An ← An − N, EA = (An)
Register indirect with offset	EA = (An) + d_{16}
Indexed register indirect with offset	EA = (An) + (Xn) + d_8
Immediate data addressing	
Immediate	DATA = next word(s)
Quick immediate	inherent data
Implied addressing	
Implied register	EA = SR, USP, SSP, PC VBR, SFC, DFC

NOTES:

EA = Effective address
An = Address register
Dn = Data register
Xn = Address or data register used as index register
SR = Status register
PC = Program counter
( ) = Contents of
d_8 = 8-bit offset (displacement)
d_{16} = 16-bit offset (displacement)
N = 1 for byte, 2 for word, and 4 for long word. If An is the stack pointer and the operand size is byte, N = 2 to keep the stack pointer on a word boundary.
← = Replaces

Instruction Set Overview

The SCN68010 instruction set is shown in Table 2. Some additional instructions are variations, or subsets, of these and they appear in Table 3. Special emphasis has been given to the instruction set's support of structured high-level languages to facilitate ease of programming. Each instruction, with few exceptions, operates on bytes, words, and long words and most instructions can use any of the 14 addressing modes. By combining instruction types, data types, and addressing modes, over 1000 useful instructions are provided. These instructions include signed and unsigned multiply and divide, "quick" arithmetic operations, BCD arithmetic, and expanded operations (through traps). Also, 33 instructions may be used in the loop mode with certain addressing modes and the DBcc instruction to provide 230 high-performance string, block manipulation, and extended arithmetic operations.

Table 2. Instruction Set

MNEMONIC	DESCRIPTION
ABCD*	Add decimal with extend
ADD*	Add
AND*	Logical AND
ASL*	Arithmetic shift left
ASR*	Arithmetic shift right
B$_{CC}$	Branch conditionally
BCHG	Bit test and change
BCLR	Bit test and clear
BRA	Branch always
BSET	Bit test and set
BSR	Branch to subroutine
BTST	Bit test
CHK	Check register against bounds
CLR*	Clear operand
CMP*	Compare
DB$_{CC}$	Decrement and branch conditionally
DIVS	Signed divide
DIVU	Unsigned divide
EOR*	Exclusive OR
EXG	Exchange registers
EXT	Sign extend
JMP	Jump
JSR	Jump to subroutine
LEA	Load effective address
LINK	Link stack
LSL*	Logical shift left
LSR*	Logical shift right
MOVE*	Move source to destination
MULS	Signed multiply
MULU	Unsigned multiply
NBCD*	Negate decimal with extend
NEG*	Negate
NOP	No operation
NOT*	One's complement
OR*	Logical OR
PEA	Push effective address
RESET	Reset external devices
ROL*	Rotate left without extend
ROR*	Rotate right without extend
ROXL*	Rotate left with extend
ROXR*	Rotate right with extend
RTD	Return and deallocate
RTE	Return from exception
RTR	Return and restore
RTS	Return from subroutine
SBCD*	Subtract decimal with extend
S$_{CC}$	Set conditional
STOP	Stop
SUB	Subtract
SWAP	Swap data register halves
TAS	Test and set operand
TRAP	Trap
TRAPV	Trap on overflow
TST*	Test
UNLK	Unlink

NOTE:
*Loopable instructions

Table 3. Variations of Instruction Types

INSTRUCTION TYPE	VARIATION	DESCRIPTION
ADD	ADD* ADDA ADDQ ADDI ADDX*	Add Add address Add quick Add immediate Add with extend
AND	AND* ANDI ANDI to CCR ANDI to SR	Logical AND AND immediate AND immediate to condition codes AND immediate to status register
CMP	CMP* CMPA* CMPM* COMPI	Compare Compare address Compare memory Compare immediate
EOR	EOR* EORI EORI to CCR EORI to SR	Exclusive OR Exclusive OR immediate Exclusive OR immediate to condition codes Exclusive OR immediate to status register
MOVE	MOVE* MOVEA* MOVEC MOVEM MOVEP MOVEQ MOVES MOVE from SR MOVE to SR MOVE from CCR MOVE to CCR MOVE USP	Move source to destination Move address Move control register Move multiple registers Move peripheral data Move quick Move alternate address space Move from status register Move to status register Move from condition codes Move to condition codes Move user stack pointer
NEG	NEG* NEGX*	Negate Negate with extend
OR	OR* ORI ORI to CCR ORI to SR	Logical OR OR immediate OR immediate to condition codes OR immediate to status register
SUB	SUB* SUBA* SUBI SUBQ SUBX*	Subtract Subtract address Subtract immediate Subtract quick Subtract with extend

NOTE:
*Loopable instructions

Virtual Memory/Machine Concepts

In most systems using the SCN68010 as the central processor, only a fraction of the 16Mbyte address space will actually contain physical memory. However, by using virtual memory techniques the system can be made to appear to the user to have 16Mbytes of physical memory available to him/her. These techniques have been used for several years in large mainframe computers and more recently in minicomputers and now, with the SCN68010, can be fully supported in microprocessor-based systems.

In a virtual memory system, a user program can be written as though it has a large amount of memory available to it when only a small amount of memory is physically present in the system. In a similar fashion, a system can be designed in such a manner as to allow user programs to access other types of devices that are not physically present in the system such as tape drives, disk drives, printers, or CRTs. With proper software emulation, a physical system can be made to appear to a user program as any other computer system and the program may be given full access to all of the resources of that emulated system. Such an emulated system is called a virtual machine.

Virtual Memory

The basic mechanisim for supporting virtual memory in computers is to provide only a limited amount of high-speed physical memory that can be accessed directly by the processor while maintaining an image of a much larger "virtual" memory on secondary storage devices such as large capacity disk drives. When the processor attempts to access a location in the virtual memory map that is not currently residing in physical memory (referred to as a page fault), the access to that location is temporarily suspended while the necessary data is fetched from the secondary storage and placed in physical memory; the suspended access is then completed. The SCN68010 provides hardware support for virtual memory with the capability of suspending an instruction's execution when a bus error is signaled and then completing the instruction after the physical memory has been updated as necessary.

The SCN68010 uses instruction continuation rather than instruction restart to support virtual memory. With instruction restart, the processor must remember the exact state of the system before each instruction is started in order to restore that state if a page fault occurs during its execution. Then, after the page fault has been repaired, the entire instruction that caused the fault is reexecuted. With instruction continuation, when a page fault occurs the processor stores its internal state and then, after the page fault is repaired, restores that internal state and continues execution of the instruction. In order for the SCN68010 to utilize instruction continuation, it stores its internal state on the supervisor stack when a bus cycle is terminated with a bus error signal. It then loads the program counter from vector table entry number two (offset $008) and resumes program execution at that new address. When the bus error exception handler routine has completed execution, an RTE instruction is executed which reloads the SCN68010 with the internal state stored on the stack, reruns the faulted bus cycle, and continues the suspended instruction. Instruction continuation has the additional advantage of allowing hardware support for virtual I/O devices. Since virtual registers may be simulated in the memory map, an access to such a register will cause a fault and the function of the register can be emulated by software.

Virtual Machine

One typical use for a virtual machine system is in the development of software such as an operating system for another machine with hardware also under development and not available for programming use. In such a system, the governing operating system (OS) emulates the hardware of the new system and allows the new OS to be executed and debugged as though it were running on the new hardware. Since the new OS is controlled by the governing OS, the new one must execute at a lower privilege level than the governing OS so that any attempts by the new OS to use virtual resources that are not physically present, and should be emulated, will be trapped by the governing OS and handled in software. In the SCN68010, a virtual machine may be fully supported by running the new OS in the user mode and the governing OS in the supervisor mode so that any attempts to access supervisor resources or execute privileged instructions by the new OS will cause a trap to the governing OS.

In order to fully support a virtual machine, the SCN68010 must protect the supervisor resources from access by user programs. The one supervisor resource that is not fully protected in the SCN68000 is the system byte of the status register. In the SCN68000, the MOVE from SR instruction allows user programs to test the S bit (in addition to the 1 bit and interrupt mask) and thus determine that they are running in the user mode. For full virtual machine support, a new OS must not be aware of the fact that it is running in the user mode and thus should not be allowed to access the S bit. For this reason, the MOVE from SR instruction on the SCN68010 is a privileged instruction and the MOVE from CCR instruction has been added to allow user programs unhindered access to the condition codes. By making the MOVE from SR instruction privileged, when the new OS attempts to access the S bit, a trap to the governing OS will occur and the SR image passed to the new OS by the governing OS will have the S bit set.

DATA ORGANIZATION AND ADDRESSING CAPABILITIES

This section contains a description of the registers and the data organization of the SCN68010.

Operand Size

Operand sizes are defined as follows: a byte equals 8 bits, a word equals 16 bits, and a long word equals 32 bits. The operand size for each instruction is either explicitly encoded in the instruction or implicitly defined by the instruction operation. Implicit instructions support some subset of all three sizes.

Data Organization in Registers

The eight data registers support data operands of 1, 8, 16, or 32 bits. The seven address registers and the stack pointers support address operands of 32 bits. The four control registers support data of 1, 3, 8, 16, or 32 bits depending on the register specified.

Data Registers

Each data register is 32 bits wide. Byte operands occupy the low order 8 bits, word operands the low order 16 bits, and long word operands the entire 32 bits. The least significant bit is addressed as bit zero; the most significant bit is addressed as bit 31.

When a data register is used as either a source or destination operand, only the appropriate low order portion is changed; the remaining high order portion is neither used nor changed.

Address Registers

Each address register and stack pointer is 32 bits wide and holds a full 32-bit address. Address registers do not support the sized operands. Therefore, when an address register is used as a source operand, either the low order word or the entire long word operand is used depending upon the operation size. When an address register is used as the destination operand, the entire register is affected regardless of the operation size. If the operation size is word, any other operands are sign extended to 32 bits before the operation is performed.

Control Registers

The status register (SR) is 16 bits wide with the lower byte being accessed as the condition code register (CCR). Not all 16 bits of the SR are defined and will be read as zeroes and ignored when written. Operations to the CCR are word operations; however, the upper byte will be read as all zeros and ignored when written.

The vector base register (VBR) is 32 bits wide and holds a full 32-bit address. All operations involving the VBR are long word operations regardless of whether it is the source or destination operand.

The alternate function code registers (SFC and DFC) are three bits wide and contain the function code values placed on FC0 - FC2 during the operand read or write of a MOVES instruction. All transfers to or from the alternate function code registers are 32 bits although the upper 29 bits will be read as zeroes and ignored when written.

Data Organization in Memory

The data types supported by the SCN68010 are: bit data, integer data of 8, 16, or 32 bits, 32-bit addresses and binary coded decimal data. Each of these data types is put in memory, as shown in Figure 4. The numbers indicate the order in which the data would be accessed from the processor.

Bytes are individually addressable with the high order byte having an even address the same as the word, as shown in Figure 5. The low order byte has an odd address that is one count higher than the word address. Instructions and word or long word data are accessed only on word (even byte) boundaries. If a long word datum is located at address n (n even), then the low-order word of that datum is located at address n + 2.

Addressing

Instructions for the SCN68010 contain two kinds of information: the type of function to be performed and the location of the operand(s) on which to perform that function. The methods used to locate (address) the operand(s) are explained in the following paragraphs.

Instructions specify an operand location in one of three ways:

Register Specification - the number of the register is given in the register field of their instruction.

Effective Address - use of the different effective addressing modes.

Implicit Reference - the definition of certain instructions implies the use of specific registers.

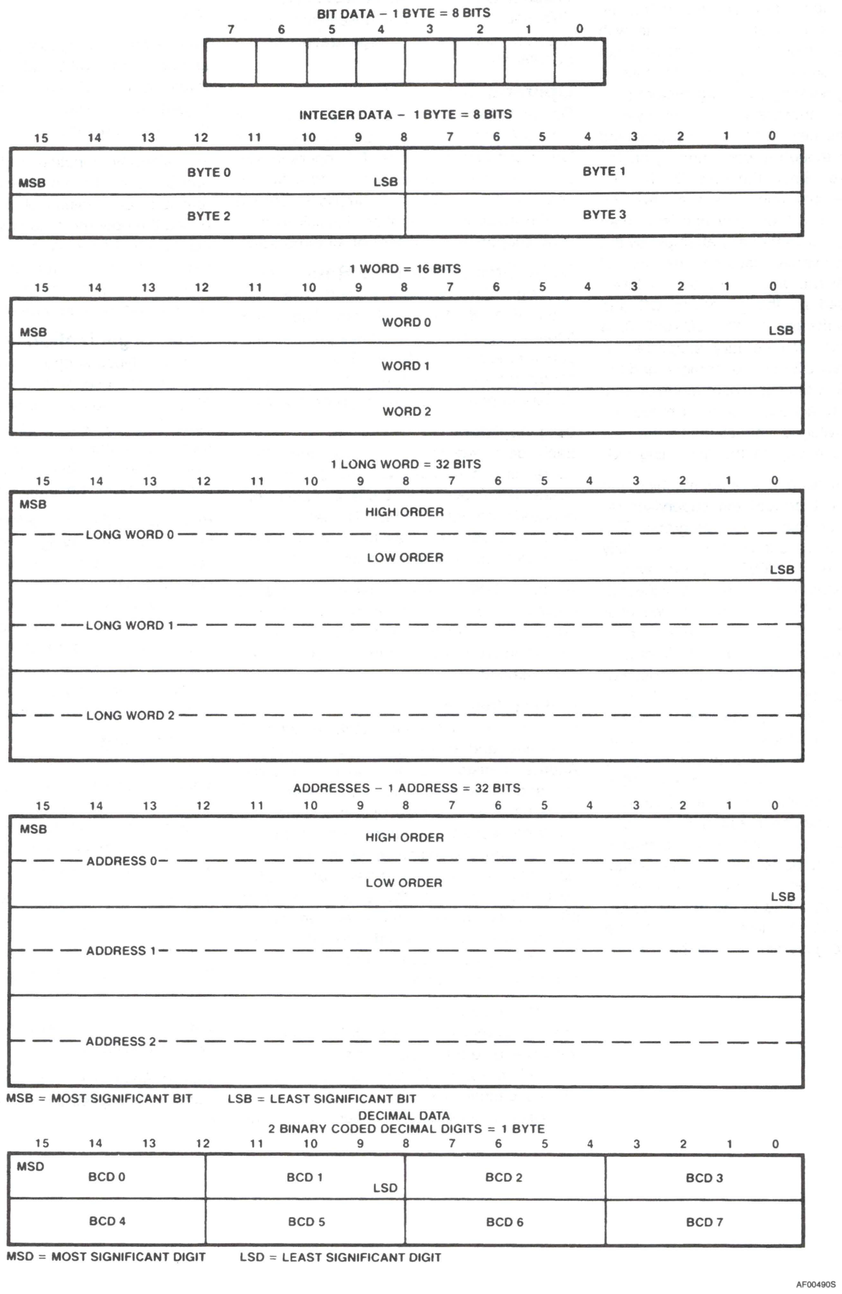

BIT DATA – 1 BYTE = 8 BITS
7 6 5 4 3 2 1 0
INTEGER DATA – 1 BYTE = 8 BITS
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
BYTE 0
LSB
BYTE 1
BYTE 2
BYTE 3
1 WORD = 16 BITS
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
WORD 0
LSB
WORD 1
WORD 2
1 LONG WORD = 32 BITS
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
HIGH ORDER
LONG WORD 0
LOW ORDER
LSB
LONG WORD 1
LONG WORD 2
ADDRESSES – 1 ADDRESS = 32 BITS
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSB
HIGH ORDER
ADDRESS 0
LOW ORDER
LSB
ADDRESS 1
ADDRESS 2
MSB = MOST SIGNIFICANT BIT
LSB = LEAST SIGNIFICANT BIT
DECIMAL DATA
2 BINARY CODED DECIMAL DIGITS = 1 BYTE
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
MSD
BCD 0
BCD 1
LSD
BCD 2
BCD 3
BCD 4
BCD 5
BCD 6
BCD 7
MSD = MOST SIGNIFICANT DIGIT
LSD = LEAST SIGNIFICANT DIGIT
AF00490S

Figure 4. Memory Data Organization

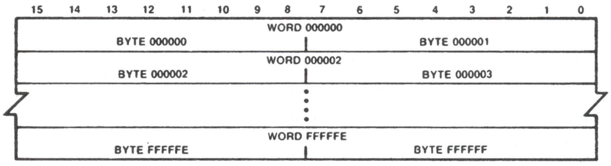

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
WORD 000000
BYTE 000000
BYTE 000001
WORD 000002
BYTE 000002
BYTE 000003
WORD FFFFFE
BYTE FFFFFE
BYTE FFFFFF
AF00500S

Figure 5. Word Organization in Memory

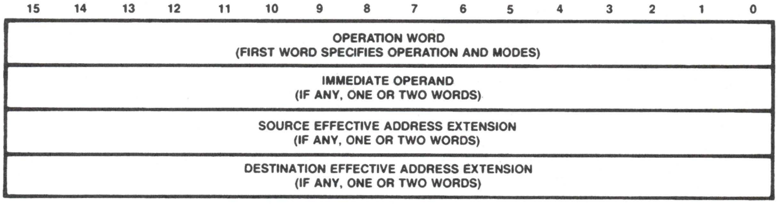

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
OPERATION WORD
(FIRST WORD SPECIFIES OPERATION AND MODES)
IMMEDIATE OPERAND
(IF ANY, ONE OR TWO WORDS)
SOURCE EFFECTIVE ADDRESS EXTENSION
(IF ANY, ONE OR TWO WORDS)
DESTINATION EFFECTIVE ADDRESS EXTENSION
(IF ANY, ONE OR TWO WORDS)
AF00510S

Figure 6. Instruction Operation Word General Format

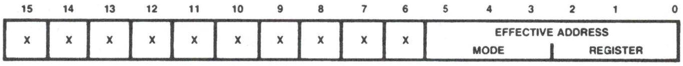

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
X X X X X X X X X X
EFFECTIVE ADDRESS
MODE
REGISTER
AF00520S

Figure 7. Single-Effective-Address Instruction Operation Word

Instruction Format

Instructions are from one to five words in length as shown in Figure 6. The length of the instruction and the operation to be performed is specified by the first word of the instruction which is called the operation word. The remaining words further specify the operands. These words are either immediate operands or extensions to the effective address mode specified in the operation word.

Program/Data References

The SCN68010 separates memory references into two classes: program references and data references. Program references, as the name implies, are references to that section of memory that contains the program being executed. Data references refer to that section of memory that contains data. Generally, operand reads are from the data space. All operand writes are to the data space.

Register Specification

The register field within an instruction specifies the register to be used. Other fields within the instruction specify whether the register selected is an address or data register and how the register is to be used.

Effective Address

Most instructions specify the location of an operand by using the effective address field in the operation word. For example, Figure 7 shows the general format of the single-effective-address instruction operation word. The effective address is composed of two 3-bit fields: the mode field and the register field. The value in the mode field selects the different address modes. The register field contains the number of a register.

The effective address field may require additional information to fully specify the operand. This additional information, called the effective address extension, is contained in the following word or words and is considered part of the instruction, as shown in Figure 6. The effective address modes are grouped into three categories: register direct, memory addressing, and special.

Register Direct Modes

These effective addressing modes specify that the operand is in one of sixteen general purpose registers or one of four control registers.

Data Register Direct — The operand is in the data register specified by the effective address register field.

Address Register Direct — The operand is in the address register specified by the effective address register field.

Memory Address Modes

These effective addressing modes specify that the operand is in memory and provide the specific address of the operand.

Address Register Indirect — The address of the operand is in the address register specified by the register field. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Address Register Indirect with Post Increment — The address of the operand is in the address register specified by the register field. After the operand address is used, it is incremented by one, two, or four depending upon whether the size of the operand is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is incremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

Address Register Indirect with Predecrement — The address of the operand is in the address register specified by the register field. Before the operand address is used, it is decremented by one, two, or four depending upon whether the operand size is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is decremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

Address Register Indirect with Displacement — This addressing mode requires one word of extension. The address of the operand is the sum of the address in the address register and the sign-extended 16-bit displacement integer in the extension word. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Address Register Indirect with Index — This addressing mode requires one word of extension. The address of the operand is the sum of the address in the address register, the sign-extended displacement integer in the low order eight bits of the extension word, and the contents of the index register. The index may be specified as the sign extended low-order word or the long word in the index register. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Special Address Modes

The special address modes use the effective address register field to specify the special addressing mode instead of a register number.

Absolute Short Address — This addressing mode requires one word of extension. The address of the operand is in the extension word. The 16-bit address is sign extended before it is used. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Absolute Long Address — This addressing mode requires two words of extension. The address of the operand is developed by the concatenation of the extension words. The high order part of the address is the first extension word; the low order part of the address is the second extension word. The reference is classified as a data reference with the exception of the jump and jump-to-subroutine instructions.

Program Counter with Displacement — This addressing mode requires one word of extension. The address of the operand is the sum of the address in the program counter and the sign-extended 16-bit displacement integer in the extension word. The value in the program counter is the address of the extension word. The reference is classified as a program reference.

Table 4. Effective Address Encoding Summary

ADDRESSING MODE	MODE	REGISTER
Data register direct	000	Register number
Address register direct	001	Register number
Address register indirect	010	Register number
Address register indirect with postincrement	011	Register number
Address register indirect with predecrement	100	Register number
Address register indirect with displacement	101	Register number
Address register indirect with index	110	Register number
Absolute short	111	000
Absolute long	111	001
Program counter with displacement	111	010
Program counter with index	111	011
Immediate	111	100

Program Counter with Index — This addressing mode requires one word of extension. The address is the sum of the address in the program counter, the sign-extended displacement integer in the lower eight bits of the extension word, and the contents of the index register. The index may be specified as the sign extended low-order word or the long word in the index register. The value in the program counter is the address of the extension word. The reference is classified as a program reference.

Immediate Data — This addressing mode requires either one or two words of extension depending on the size of the operation.

Byte Operation	operand is in the low order byte of extension word
Word Operation	operand is in the extension word
Long Word Operation	operand is in the two extension words, high order 16 bits are in the first extension word, low order 16 bits are in the second extension word.

Implicit Reference — Some instructions make implicit reference to the program counter (PC), the system stack pointer (SP), the supervisor stack pointer (SSP), the user stack pointer (USP), the status register (SR), the condition code register (CCR), the vector base register (VBR), or the alternate function code registers (SFC or DFC).

A selected set of instructions may reference the status register by means of the effective address field. These are:

ANDI to CCR
ANDI to SR
EORI to CCR
EORI to SR
ORI to CCR
ORI to SR
MOVE to CCR
MOVE to SR
MOVE from SR

Effective Address Encoding Summary

Table 4 is a summary of the effective addressing modes discussed in the previous paragraphs.

System Stack

The system stack is used implicitly by many instructions; user stacks and queues may be created and maintained through the addressing modes. Address register seven (A7) is the system stack pointer (SP). The system stack pointer is either the supervisor stack pointer (SSP) or the user stack pointer (USP), depending on the state of the S bit in the status register. If the S bit indicates supervisor state, the SSP is the active system stack pointer and the USP cannot be referenced as an address register. If the S bit indicates user state, the USP is the active system stack pointer, and the SSP cannot be referenced. Each system stack fills from high memory to low memory.

INSTRUCTION SET SUMMARY

This section contains an overview of the form and structure of the SCN68010 instruction set. The instructions form a set of tools that include all the machine functions to perform the following operations:

Data movement
Integer arithmetic
Logical
Shift and rotate
Bit manipulation
Binary code decimal
Program control
System control

The complete range of instruction capabilities combined with the flexible addressing modes described previously provide a very flexible base for program development.

Data Movement Operations

The basic method of data acquisition (transfer and storage) is provided by the move (MOVE) instruction. The move instruction and the effective addressing modes allow both address and data manipulation. Data movement instructions allow byte, word, and long word operands to be transferred from memory to memory, memory to register, register to memory, and register to register. Address movement instructions allow word and long word operand transfers and ensure that only legal address manipulations are executed. In addition to the general move instruction there are several special data movement instructions: move multiple registers (MOVEM), move peripheral data (MOVEP), exchange registers (EXG), load effective address (LEA), push effective address (PEA), link stack (LINK), unlink stack (UNLK), move quick (MOVEQ), move control register (MOVECS), and move alternate address space (MOVES). Table 5 is a summary of the data movement operations.

Integer Arithmetic Operations

The arithmetic operations include the four basic operations of add (ADD), subtract (SUB), multiply (MUL), and divide (DIV) as well as arithmetic compare (CMP), clear (CLR), and negate (NEG). The add and subtract instructions are available for both address and data operations, with data operations accepting all operand sizes. Address operations are limited to legal address size operands (16 or 32 bits). Data, address, and memory compare operations are also available. The clear and negate instructions may be used on all sizes of data operands.

The multiply and divide operations are available for signed and unsigned operands using word multiply to produce a long word product, and a long word dividend with word divisor to produce a word quotient with a word remainder.

Multiprecision and mixed size arithmetic can be accomplished using a set of extended instructions. These instructions are: add extended (ADDX), subtract extended (SUBX), sign extend (EXT), and negate binary with extend (NEGX).

Table 5. Data Movement Operations

INSTRUCTION	OPERAND SIZE	OPERATION
EXG	32	Rx ↔ Ry
LEA	32	EA → An
LINK	–	(An) → –(SP) (SP) → An (SP) + displacement → SP
MOVE	8, 16, 32	(EA)s → EAd
MOVEC	32	(Rn) → Cr (Cr) → Rn
MOVEM	16, 32	(EA) → An, Dn An, Dn → EA
MOVES	8, 16, 32	(EA) → Rn (Rn) → EA
MOVEP	16, 32	d(An) → Dn Dn → d(An)
MOVEQ	8	#xxx → Dn
PEA	32	EA → –(SP)
SWAP	32	Dn[31:16] ↔ Dn[15:0]
UNLK	–	An → Sp (SP) + → An

NOTES:
s = source
d = destination
[ ] = bit numbers
–( ) = indirect with predecrement
( ) + = indirect with postdecrement
= immediate data

Table 6. Integer Arithmetic Operations

INSTRUCTION	OPERAND SIZE	OPERATION
ADD	8, 16, 32 16, 32	(Dn) + (EA) → Dn (EA) + (Dn) → EA (EA) + #xxx → EA (An) + (EA) → An
ADDX	8, 16, 32 16, 32	(Dx) + (Dy) + X → Dx –(Ax) + –(Ay) + X → (Ax)
CLR	8, 16, 32	0 → (EA)
CMP	8, 16, 32 16, 32	(Dn) – (EA) (EA) – #xxx (Ax) + –(Ay) + (An) – (EA)
DIVS	32÷16	(Dn)/(EA) → Dn
DIVU	32÷16	(Dn)/(EA) → Dn
EXT	8 → 16 16 → 32	$(Dn)_8 \rightarrow Dn_{16}$ $(Dn)_{16} \rightarrow Dn_{32}$
MULS	16 × 16 → 32	(Dn) × (EA) → Dn
MULU	16 × 16 → 32	(Dn) × (EA) → Dn
NEG	8, 16, 32	0 – (EA) → EA
NEGX	8, 16, 32	0 – (EA) – X → EA
SUB	8, 16, 32 16, 32	(Dn) – (EA) → Dn (EA) – Dn → EA (EA) – #xxx → EA (An) – (EA) → An
SUBX	8, 16, 32	(Dx) – (Dy) – X → Dx –(Ax) – –(Ay) – X → (Ax)
TAS	8	[EA] – 0, 1 → EA[7]
TST	8, 16, 32	(EA) – 0

NOTES:
[ ] = bit number
= immediate data
–( ) = indirect with predecrement
( )+ = indirect with postdecrement

A test operand (TST) instruction that will set the condition codes as a result of a compare of the operand with zero is also available. Test and set (TAS) is a synchronization instruction useful in multiprocessor systems. Table 6 is a summary of the integer arithmetic operations.

Logical Operations

Logical operation instructions AND, OR, EOR, and NOT are available for all sizes of integer data operands. A similar set of immediate instructions (ANDI, ORI, and EORI) provide these logical operations with all sizes of immediate data. Table 7 is a summary of the logical operations.

Table 7. Logical Operations

INSTRUCTION	OPERAND SIZE	OPERATION
AND	8, 16, 32	(Dn) ∧ (EA) → Dn (EA) ∧ (Dn) → EA (EA) ∧ #xxx → EA
OR	8, 16, 32	(Dn) v (EA) → Dn (EA) v (Dn) → EA (EA) v #xxx → EA
EOR	8, 16, 32	(EA) ⊕ (Dy) → EA (EA) ⊕ #xxx → EA
NOT	8, 16, 32	~(EA) → EA

NOTES:
= immediate data
~ = invert
∧ = logical AND
v = logical OR
⊕ = logical exclusive OR

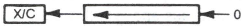
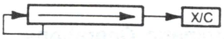
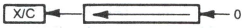
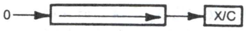
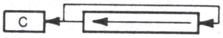
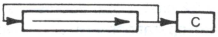
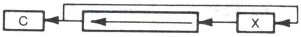
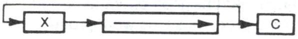
Shift and Rotate Operations

Shift operations in both directions are provided by the arithmetic shift instructions ASR and ASL and logical shift instructions LSR and LSL. The rotate instructions (with and without extend) available are ROXR, ROXL, ROR, and ROL. All shift and rotate operations can be performed in either registers or memory. Register shifts and rotates support all operand sizes and allow a shift count specified in a data register.

Memory shifts and rotates are for word operands only and allow only single-bit shifts or rotates.

Table 8 is a summary of the shift and rotate operations.

Table 8. Shift and Rotate Operations

INSTRUCTION	OPERAND SIZE	OPERATION
ASL	8, 16, 32	
ASR	8, 16, 32	
LSL	8, 16, 32	
LSR	8, 16, 32	
ROL	8, 16, 32	
ROR	8, 16, 32	
ROXL	8, 16, 32	
ROXR	8, 16, 32	

Bit Manipulation Operations

Bit manipulation operations are accomplished using the following instructions: bit test (BTST), bit test and set (BSET), bit test and clear (BCLR), and bit test and change (BCHG). Table 9 is a summary of the bit manipulation operations. (Z is bit 2 of the status register.)

Table 9. Bit Manipulation Operations

INSTRUCTION	OPERAND SIZE	OPERATION
BTST	8, 32	~bit of (EA) → Z
BSET	8, 32	~bit of (EA) → Z 1 → bit of EA
BCLR	8, 32	~bit of (EA) → Z 0 → bit of EA
BCHG	8, 32	~bit of (EA) → Z ~bit of (EA) → bit of EA

NOTE:
~ = invert

Binary Coded Decimal Operations

Multiprecision arithmetic operations on binary coded decimal numbers are accomplished using the following instructions: add decimal with extend (ABCD), subtract decimal with extend (SBCD), and negate decimal with extend (NBCD). Table 10 is a summary of the binary coded decimal-operations.

Table 10. Binary Coded Decimal Operations

INSTRUCTION	OPERAND SIZE	OPERATION
ABCD	8	$(Dx)_{10} + (Dy)_{10} + X \rightarrow Dx$ $-(Ax)_{10} + -(Ay)_{10} + X \rightarrow (Ax)$
SBCD	8	$(Dx)_{10} - (Dy)_{10} - X \rightarrow Dx$ $-(Ax)_{10} - -(Ay)_{10} - X \rightarrow (Ax)$
NBCD	8	$0 - (EA)_{10} - X \rightarrow (EA)$

NOTES:
− = indirect with predecrement.
+ = indirect with postdecrement.

Program Control Operations

Program control operations are accomplished using a series of conditional and unconditional branch instructions and return instructions. These instructions are summarized in Table 11.

The conditional instructions provide setting and branching for the following conditions:

CC — carry clear
CS — carry set
EQ — equal
F — never true
GE — greater or equal
GT — greater than
HI — high
LE — less or equal
LS — low or same
LT — less than
MI — minus
NE — not equal
PL — plus
T — always true
VC — no overflow
VS — overflow

Table 11. Program Control Operations

INSTRUCTION	OPERATION
Conditional	
B_{CC}	Branch conditionally (14 conditions) 8- and 16-bit displacement
DB_{CC}	Test condition, decrement, and branch 16-bit displacement
S_{CC}	Set byte conditionally (16 conditions)
Unconditional	
BRA	Branch always 8- and 16-bit displacement
BSR	Branch to subroutine 8- and 16-bit displacement
JMP	Jump
JSR	Jump to subroutine
Returns	
RTD	Return from subroutine and deallocate stack
RTR	Return and restore condition codes
RTS	Return from subroutine

System Control Operations

System control operations are accomplished by using privileged instructions, trap generating instructions, and instructions that use or modify the condition code register. These instructions are summarized in Table 12.

SIGNAL AND BUS OPERATION DESCRIPTION

This section contains a brief description of the input and output signals. A discussion of bus operation during the various machine cycles and operations is also given.

NOTE

The terms **assertion** and **negation** will be used extensively. This is done to avoid confusion when dealing with a mixture of "active-low" and "active-high" signals. The term assert or assertion is used to indicate that a signal is active or true, independent of whether that level is represented by a high or low voltage. The term negate or negation is used to indicate that a signal is inactive or false.

Signal Description

The input and output signals can be functionally organized into the groups shown in Figure 8. The following paragraphs provide a brief description of the signals and a reference (if applicable) to other paragraphs that contain more detail about the function being performed.

Table 12. System Control Operations

INSTRUCTION	OPERATION
Privileged	
ANDI to SR	Logical AND to status register
EORI to SR	Logical EOR to status register
MOVE EA to SR	Load new status register
MOVE SR to EA	Store status register
MOVE USP	Move user stack pointer
MOVEC	Move control register
MOVES	Move alternate address space
ORI to SR	Logical OR to status register
RESET	Reset external devices
RTE	Return from exception
STOP	Stop program execution
Trap Generating	
CHK	Check data register against upper bounds
TRAP	Trap
TRAPV	Trap on overflow
Condition Code Register	
ANDI to CCR	Logical AND to condition codes
EORI to CCR	Logical EOR to condition codes
MOVE EA to CCR	Load new condition codes
MOVE CCR to EA	Store condition codes
ORI to CCR	Logical OR to condition codes

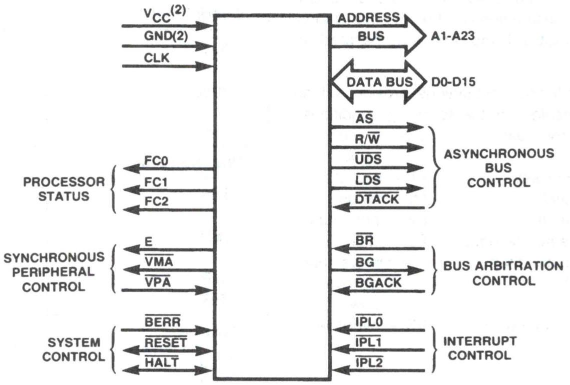

V_{CC}(2)
GND(2)
CLK
ADDRESS BUS
A1-A23
DATA BUS
D0-D15
$\overline{AS}$
R/$\overline{W}$
$\overline{UDS}$
$\overline{LDS}$
$\overline{DTACK}$
ASYNCHRONOUS BUS CONTROL
PROCESSOR STATUS
FC0
FC1
FC2
SYNCHRONOUS PERIPHERAL CONTROL
E
$\overline{VMA}$
$\overline{VPA}$
$\overline{BR}$
$\overline{BG}$
$\overline{BGACK}$
BUS ARBITRATION CONTROL
SYSTEM CONTROL
$\overline{BERR}$
$\overline{RESET}$
$\overline{HALT}$
$\overline{IPL0}$
$\overline{IPL1}$
$\overline{IPL2}$
INTERRUPT CONTROL
LS00560S

Figure 8. Input and Output Signals

Table 13. Data Strobe Control of Data Bus

UDS	LDS	R/W	D8 – D15	D0 – D7
High	High	–	No valid data	No valid data
Low	Low	High	Valid data bits 8 – 15	Valid data bits 0 – 7
High	Low	High	No valid data	Valid data bits 0 – 7
Low	High	High	Valid data bits 8 – 15	No valid data
Low	Low	Low	Valid data bits 8 – 15	Valid data bits 0 – 7
High	Low	Low	Valid data bits 0 – 7*	Valid data bits 0 – 7
Low	High	Low	Valid data bits 8 – 15	Valid data bits 8 – 15*

NOTE:
*These conditions are a result of current implementation and may not appear on future devices.

Address Bus (A1 through A23)
This 23-bit, unidirectional, 3-State bus is capable of addressing 8 megawords of data. It provides the address for bus operation during all cycles except CPU space cycles.

Data Bus (D0 through D15)
This 16-bit, bidirectional, 3-State bus is the general purpose data path. It can transmit and accept data in either word or byte length.

Asynchronous Bus Control
Asynchronous data transfers are handled using the following control signals: address strobe, read/write, upper and lower data strobes, and data transfer acknowledge. These signals are explained in the following paragraphs.

Address Strobe (AS) — This signal indicates that there is a valid address on the address bus.

Read/Write (R/W) — This signal defines the data bus transfer as a read or write cycle. The R/W signal also works in conjunction with the data strobes as explained in the following paragraph.

Upper and Lower Data Strobe (UDS, LDS) — These signals control the flow of data on the data bus, as shown in Table 13. When the R/W line is high, the processor will read from the data bus as indicated. When the R/W line is low, the processor will write to the data bus as shown.

Data Transfer Acknowledge (DTACK) — This input indicates that the data transfer is completed. When the processor recognizes DTACK during a read cycle, data is latched one clock cycle later and the bus cycle terminated. When DTACK is recognized during a write cycle, the bus cycle is terminated. Refer to **Asynchronous Versus Synchrcnous Operation.**

Bus Arbitration Control
The three signals, bus request, bus grant, and bus grant acknowledge, form a bus arbitration circuit to determine which device will be the bus master device.

Bus Request (BR) — This input is wire ORed with all other devices that could be bus masters. This input indicates to the processor that some other device desires to become the bus master.

Bus Grant (BG) — This output indicates to all other potential bus master devices that the processor will release bus control at the end of the current bus cycle.

Bus Grant Acknowledge (BGACK) — This input indicates that some other device has become the bus master. This signal should not be asserted until the following four conditions are met:

1. a bus grant has been received,
2. address strobe is inactive which indicates that the microprocessor is not using the bus,
3. data transfer acknowledge is inactive which indicates that neither memory or peripherals are using the bus, and
4. bus grant acknowledge is inactive which indicates that no other device is still claiming bus mastership.

Interrupt Control (IPL0, IPL1, IPL2)
These input pins indicate the encoded priority level of the device requesting an interrupt. Level seven is the highest priority while level zero indicates that no interrupts are requested. Level seven cannot be masked. The least significant bit is IPL0 and the most significant bit is IPL2. These lines must remain stable until the processor signals interrupt acknowledge (FC0 – FC2 are all high, A4 – A23 are all high) to insure that the interrupt is recognized.

System Control
The system control inputs are used to either reset or halt the processor and to indicate to the processor that bus errors have occurred. The three system control inputs are explained in the following paragraphs.

Bus Error (BERR) — This input informs the processor that there is a problem with the cycle currently being executed. Problems may be a result of:

1. nonresponding devices,
2. interrupt vector number acquisition failure,
3. illegal access request as determined by a memory management unit, or
4. other application dependent errors.

The bus error signal interacts with the halt signal to determine if the current bus cycle should be reexecuted or if exception processing should be performed. Refer to **Bus Error and Halt Operation** for additional information about the interaction of the BERR and HALT signals.

Reset (RESET) — This bidirectional signal line acts to reset (start a system initialization sequence) the processor in response to an external reset signal. An internally generated reset (result of a reset instruction) causes all external devices to be reset and the internal state of the processor is not affected. A total system reset (processor and external devices) is the result of external HALT and RESET signals applied at the same time. Refer to **Reset Operation** for further information.

Halt (HALT) — when this bidirectional line is driven by an external device, it will cause the processor to stop at the completion of the current bus cycle. When the processor has been halted using this input, all control signals are inactive and all 3-State lines are put in their high-impedance state (refer to Table 15). Refer to **Bus Error and Halt Operation** for additional information about the interaction between the HALT and BERR signals.

When the processor has stopped executing instructions, due to a double bus fault condition (refer to **Double Bus Faults**), the HALT line is driven by the processor to indicate to external devices that the processor has stopped.

Peripheral Control
These control signals are used to allow the interfacing of synchronous peripheral devices with the asynchronous SCN68010. These signals are explained in the following paragraphs.

Enable (E) — This signal is the standard enable signal common to all synchronous type peripheral devices. The period for this output is ten SCN68010 clock periods (six clocks low, four clocks high). Enable is generated by an internal ring counter which may come up in any state (i.e., at power on, it is impossible to guarantee phase relationship of E to CLK). E is a free-running clock and runs regardless of the state of the bus on the MPU.

Valid Peripheral Address ($\overline{\text{VPA}}$) — This input indicates that the device addressed is a synchronous family device and that data transfer should be synchronized with the enable (E) signal. This input also indicates that the processor should use automatic vectoring for an interrupt. Refer to **Interface with Synchronous Peripherals.**

Valid Memory Address ($\overline{\text{VMA}}$) — This output is used to indicate to synchronous peripheral devices that there is a valid address on the address bus and the processor is synchronized to enable (E). This signal only responds to a valid peripheral address ($\overline{\text{VPA}}$) input which indicates that the peripheral is a synchronous family device.

Processor Status (FC0, FC1, FC2). These function code outputs indicate the state (user or supervisor) and the address space currently being accessed, as shown in Table 14. The information indicated by the function code outputs is valid whenever address strobe ($\overline{\text{AS}}$) is active.

Clock (CLK)

The clock input is a TTL-compatible signal that is internally buffered for development of the internal clocks needed by the processor. The clock input should not be gated off at any time and the clock signal must conform to minimum and maximum pulse width times.

Signal Summary

Table 15 is a summary of all the signals discussed in the previous paragraphs.

Table 14. Function Code Outputs

FUNCTION CODE OUTPUT			CYCLE TYPE
FC2	FC1	FC0	
0	0	0	Undefined, reserved*
0	0	1	User data space
0	1	0	User program space
0	1	1	Undefined, reserved*
1	0	0	Undefined, reserved*
1	0	1	Supervisor data space
1	1	0	Supervisor program space
1	1	1	CPU space

NOTE:
*Address space 3 is reserved for user definition, while 0 and 4 are reserved for future use by Signetics.

Table 15. Signal Summary

SIGNAL NAME	MNEMONIC	INPUT/OUTPUT	ACTIVE STATE	Hi-Z	
				on $\overline{\text{HALT}}$	on $\overline{\text{BGACK}}$
Address bus	A1 – A23	Output	High	Yes	Yes
Data bus	D0 – D15	Input/output	High	Yes	Yes
Address strobe	$\overline{\text{AS}}$	Output	Low	No	Yes
Read/write	R/$\overline{\text{W}}$	Output	Read-High Write-Low	No No	Yes Yes
Upper and lower data strobes	$\overline{\text{UDS}}$, $\overline{\text{LDS}}$	Output	Low	No	Yes
Data transfer acknowledge	$\overline{\text{DTACK}}$	Input	Low	–	–
Bus request	$\overline{\text{BR}}$	Input	Low	–	–
Bus grant	$\overline{\text{BG}}$	Output	Low	No	No
Bus grant acknowledge	$\overline{\text{BGACK}}$	Input	Low	–	–
Interrupt priority level	$\overline{\text{IPL0}}$, $\overline{\text{IPL1}}$, $\overline{\text{IPL2}}$	Input	Low	–	–
Bus error	$\overline{\text{BERR}}$	Input	Low	–	–
Reset	$\overline{\text{RESET}}$	Input/Output	Low	No*	No*
Halt	$\overline{\text{HALT}}$	Input/Output	Low	No*	No*
Enable	E	Output	High	No	No
Valid memory address	$\overline{\text{VMA}}$	Output	Low	No	Yes
Valid peripheral address	$\overline{\text{VPA}}$	Input	Low	–	–
Function code output	FC0, FC1, FC2	Output	High	No	Yes
Clock	CLK	Input	High	–	–
Power input	V_{CC}	Input	–	–	–
Ground	GND	Input	–	–	–

NOTE:
*Open drain

Bus Operation

The following paragraphs explain control signal and bus operation during data transfer operations, bus arbitration, bus error and halt conditions, and reset operation.

Data Transfer Operations

Transfer of data between devices involves the following signals:

1. address bus A1 through A23,
2. data bus D0 through D15, and
3. control signals.

The address and data buses are separate parallel buses used to transfer data using an asynchronous bus structure. In all cycles, the bus master assumes responsibility for deskewing all signals it issues at both the start and end of a cycle. In addition, the bus master is responsible for deskewing the acknowledge and data signals from the slave device.

The following paragraphs explain the read, write, and read-modify-write cycles. The indivisible read-modify-write cycle is the method used by the SCN68010 for interlocked multiprocessor communications.

Read Cycle — During a read cycle, the processor receives data from the memory or a peripheral device. The processor reads bytes of data in all cases. If the instruction specifies a word (or long word) operation, the processor reads both upper and lower bytes simultaneously by asserting both upper and lower data strobes. When the instruction specifies byte operation, the processor uses an internal A0 bit to determine which byte to read and then issues the data strobe required for that byte. For byte operations, when the A0 bit equals zero, the upper data strobe is issued. When the A0 bit equals one, the lower data strobe is issued. When the data is received, the processor correctly positions it internally. If $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, or $\overline{\text{VPA}}$ is not asserted for the required setup time before the falling edge of S4, a wait cycle will be inserted in the bus cycle and $\overline{\text{DTACK}}$ will be sampled again on the falling edge of each wait cycle. The SCN68010 will continue to insert wait cycles until $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, or $\overline{\text{VPA}}$ is recognized.

A word read cycle flowchart is given in Figure 9. A byte read cycle flowchart is given in Figure 10. Read cycle timing is given in Figure 11. Figure 12 details word and byte read cycle operations.

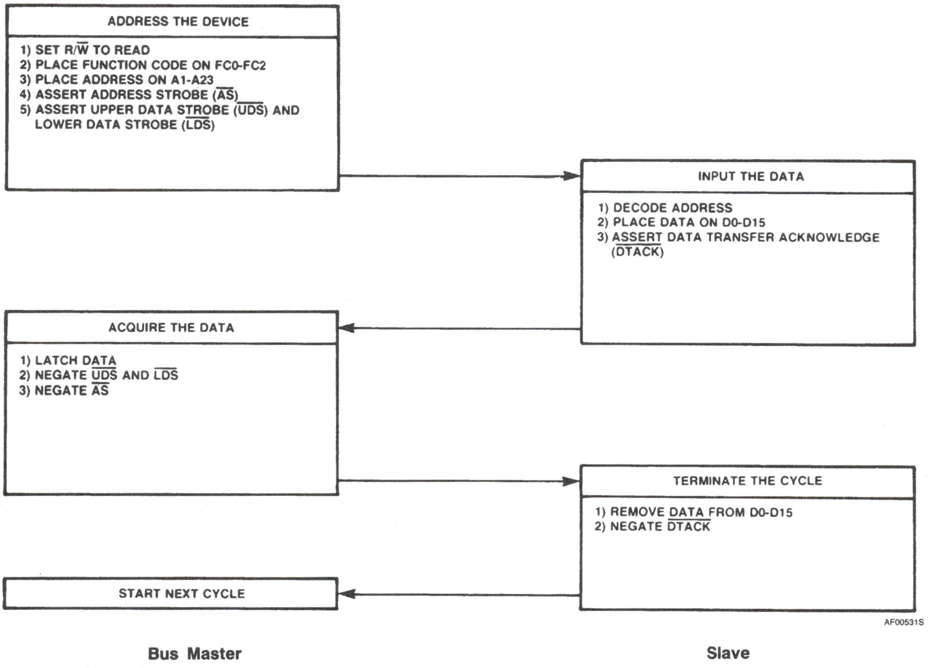
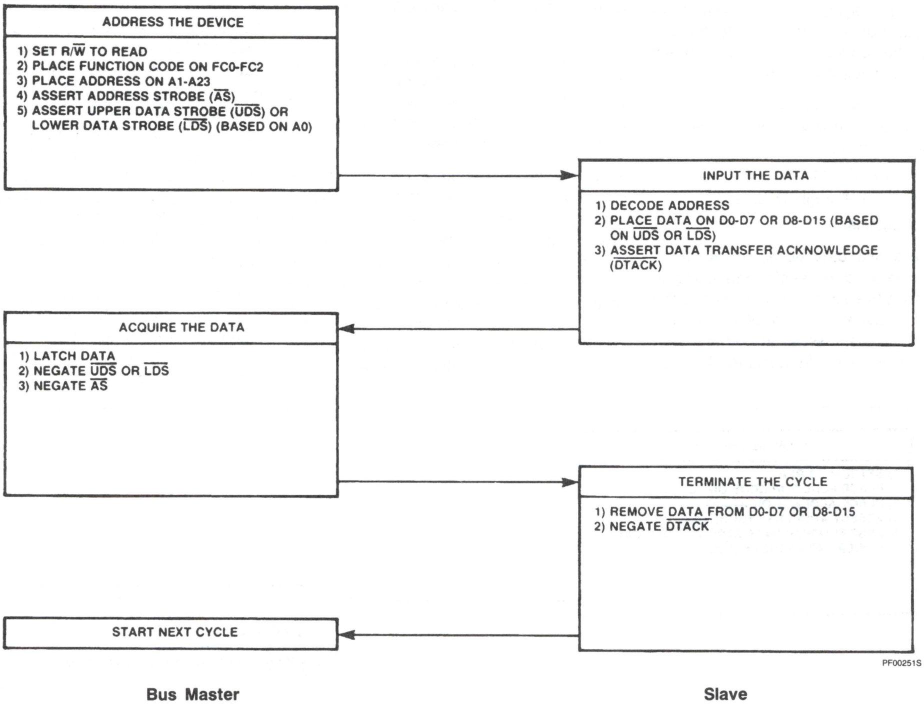

ADDRESS THE DEVICE
1) SET R/W̄ TO READ
2) PLACE FUNCTION CODE ON FC0-FC2
3) PLACE ADDRESS ON A1-A23
4) ASSERT ADDRESS STROBE (AS̄)
5) ASSERT UPPER DATA STROBE (UDS̄) AND LOWER DATA STROBE (LDS̄)
INPUT THE DATA
1) DECODE ADDRESS
2) PLACE DATA ON D0-D15
3) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK̄)
ACQUIRE THE DATA
1) LATCH DATA
2) NEGATE UDS̄ AND LDS̄
3) NEGATE AS̄
TERMINATE THE CYCLE
1) REMOVE DATA FROM D0-D15
2) NEGATE DTACK̄
START NEXT CYCLE
AF00531S
Bus Master
Slave

Figure 9. Word Read Cycle Flowchart


ADDRESS THE DEVICE
INPUT THE DATA
ACQUIRE THE DATA
TERMINATE THE CYCLE
START NEXT CYCLE
PF00251S
Bus Master
Slave

ADDRESS THE DEVICE

1) SET R/$\overline{\mathrm{W}}$ TO READ
2) PLACE FUNCTION CODE ON FC0-FC2
3) PLACE ADDRESS ON A1-A23
4) ASSERT ADDRESS STROBE ($\overline{\mathrm{AS}}$)
5) ASSERT UPPER DATA STROBE ($\overline{\mathrm{UDS}}$) OR LOWER DATA STROBE ($\overline{\mathrm{LDS}}$) (BASED ON A0)

INPUT THE DATA

1) DECODE ADDRESS
2) PLACE DATA ON D0-D7 OR D8-D15 (BASED ON $\overline{\mathrm{UDS}}$ OR $\overline{\mathrm{LDS}}$)
3) ASSERT DATA TRANSFER ACKNOWLEDGE ($\overline{\mathrm{DTACK}}$)

ACQUIRE THE DATA

1) LATCH DATA
2) NEGATE $\overline{\mathrm{UDS}}$ OR $\overline{\mathrm{LDS}}$
3) NEGATE $\overline{\mathrm{AS}}$

TERMINATE THE CYCLE

1) REMOVE DATA FROM D0-D7 OR D8-D15
2) NEGATE $\overline{\mathrm{DTACK}}$

START NEXT CYCLE

Figure 10. Byte Read Cycle Flowchart

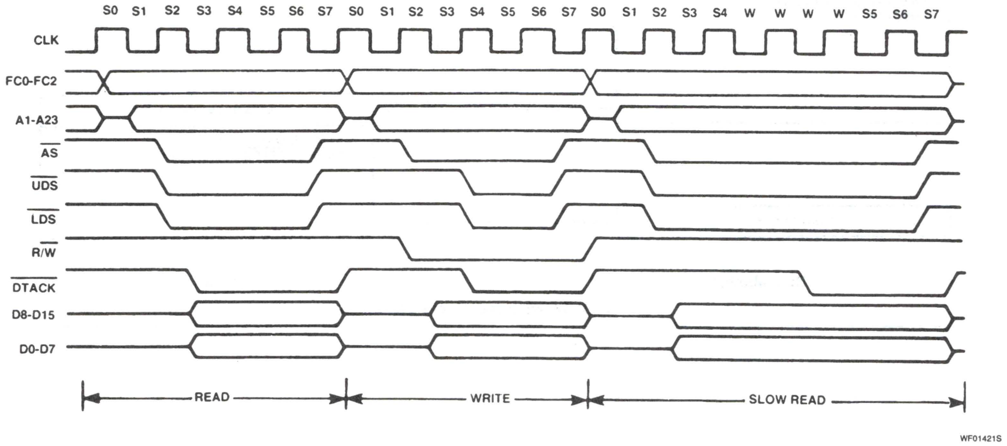

S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 W W W W S5 S6 S7
CLK
FC0-FC2
A1-A23
AS
UDS
LDS
R/W
DTACK
D8-D15
D0-D7
READ
WRITE
SLOW READ
WF01421S

Figure 11. Read and Write Cycle Timing Diagram

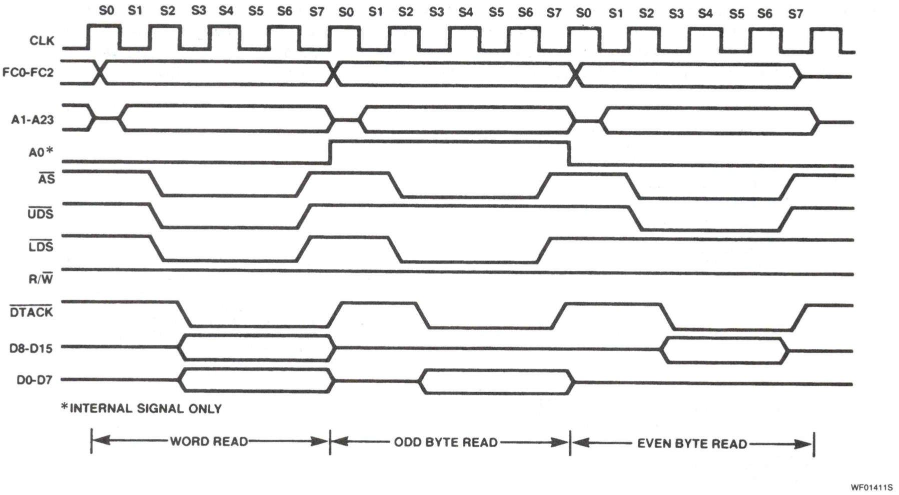

S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7
CLK
FC0-FC2
A1-A23
A0*
AS
UDS
LDS
R/W
DTACK
D8-D15
D0-D7
*INTERNAL SIGNAL ONLY
WORD READ
ODD BYTE READ
EVEN BYTE READ
WF01411S

Figure 12. Word and Byte Read Cycle Timing Diagram

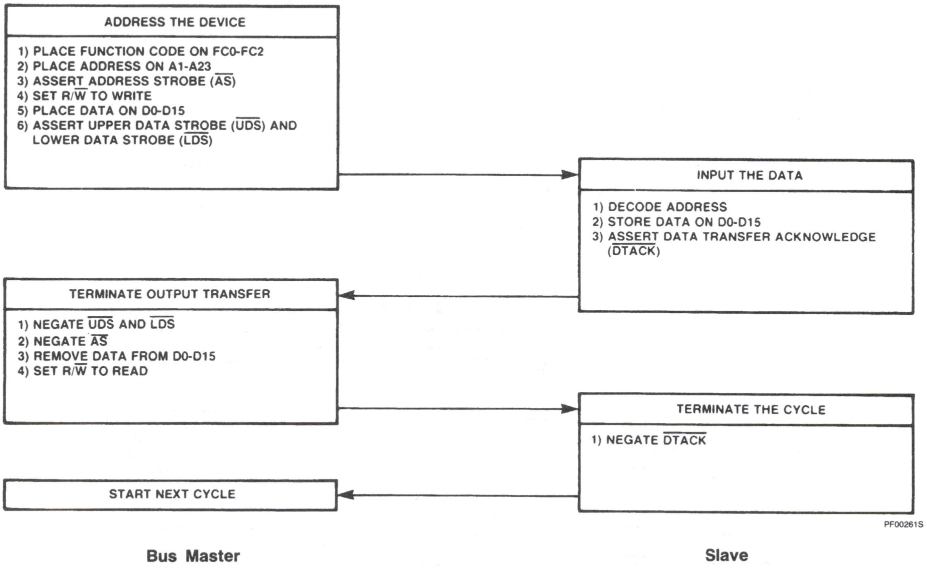

ADDRESS THE DEVICE
1) PLACE FUNCTION CODE ON FC0-FC2
2) PLACE ADDRESS ON A1-A23
3) ASSERT ADDRESS STROBE (AS)
4) SET R/W TO WRITE
5) PLACE DATA ON D0-D15
6) ASSERT UPPER DATA STROBE (UDS) AND LOWER DATA STROBE (LDS)
INPUT THE DATA
1) DECODE ADDRESS
2) STORE DATA ON D0-D15
3) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
TERMINATE OUTPUT TRANSFER
1) NEGATE UDS AND LDS
2) NEGATE AS
3) REMOVE DATA FROM D0-D15
4) SET R/W TO READ
TERMINATE THE CYCLE
1) NEGATE DTACK
START NEXT CYCLE
PF00261S
Bus Master
Slave

Figure 13. Word Write Cycle Flowchart

16-Bit Virtual Memory Microprocessor SCN68010

Write Cycle — During a write cycle, the processor sends data to either the memory or a peripheral device. The processor writes bytes of data in all cases. If the instruction specifies a word operation, the processor writes both bytes. When the instruction specifies a byte operation, the processor uses an internal A0 bit to determine which byte to write and then issues the data strobe required for that byte. For byte operations, when the A0 bit equals zero, the upper data strobe is issued. When the A0 bit equals one, the lower data strobe is issued. A word write flowchart is given in Figure 13. A byte write cycle flowchart is given in Figure 14. Write cycle timing is given in Figure 11. Figure 15 details word and byte write cycle operation.

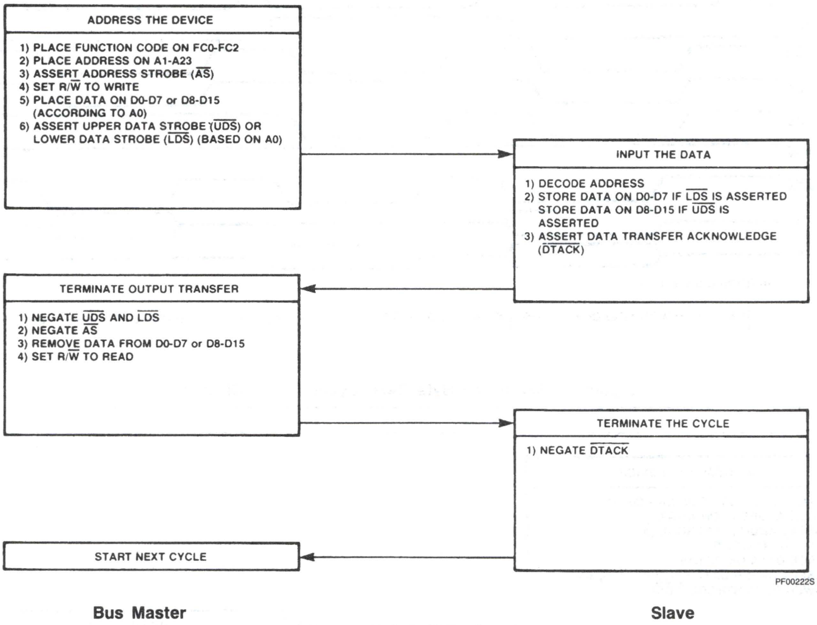

ADDRESS THE DEVICE
1) PLACE FUNCTION CODE ON FC0-FC2
2) PLACE ADDRESS ON A1-A23
3) ASSERT ADDRESS STROBE (AS)
4) SET R/W TO WRITE
5) PLACE DATA ON D0-D7 or D8-D15 (ACCORDING TO A0)
6) ASSERT UPPER DATA STROBE (UDS) OR LOWER DATA STROBE (LDS) (BASED ON A0)
INPUT THE DATA
1) DECODE ADDRESS
2) STORE DATA ON D0-D7 IF LDS IS ASSERTED STORE DATA ON D8-D15 IF UDS IS ASSERTED
3) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
TERMINATE OUTPUT TRANSFER
1) NEGATE UDS AND LDS
2) NEGATE AS
3) REMOVE DATA FROM D0-D7 or D8-D15
4) SET R/W TO READ
TERMINATE THE CYCLE
1) NEGATE DTACK
START NEXT CYCLE
PF00222S
Bus Master
Slave

Figure 14. Byte Write Cycle Flowchart

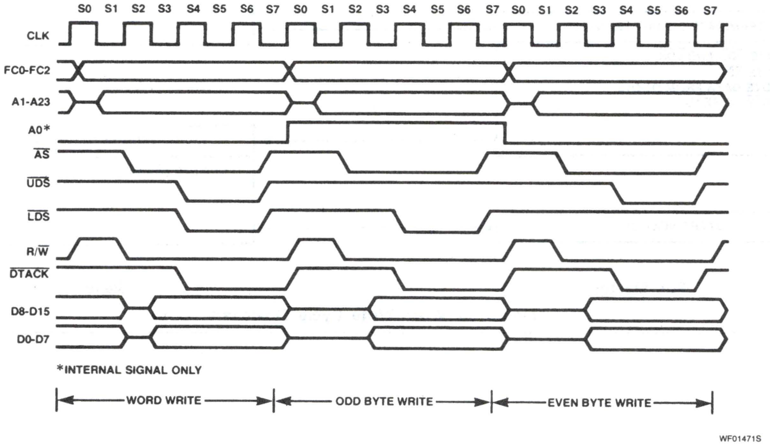

S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7 S0 S1 S2 S3 S4 S5 S6 S7
CLK
FC0-FC2
A1-A23
A0*
AS
UDS
LDS
R/W
DTACK
D8-D15
D0-D7
*INTERNAL SIGNAL ONLY
WORD WRITE
ODD BYTE WRITE
EVEN BYTE WRITE
WF01471S

Figure 15. Word and Byte Write Cycle Timing Diagram

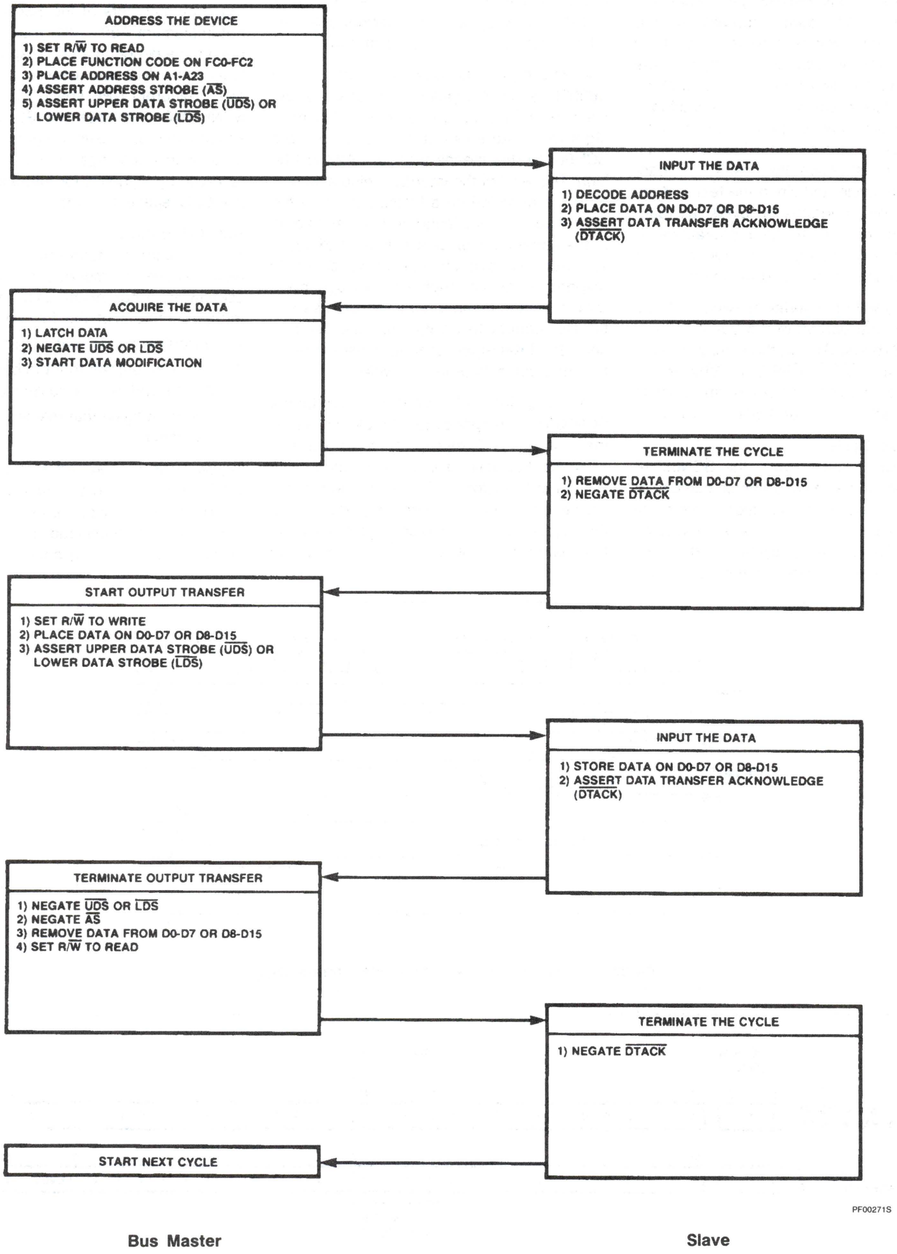

ADDRESS THE DEVICE
1) SET R/W TO READ
2) PLACE FUNCTION CODE ON FC0-FC2
3) PLACE ADDRESS ON A1-A23
4) ASSERT ADDRESS STROBE (AS)
5) ASSERT UPPER DATA STROBE (UDS) OR LOWER DATA STROBE (LDS)
INPUT THE DATA
1) DECODE ADDRESS
2) PLACE DATA ON D0-D7 OR D8-D15
3) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
ACQUIRE THE DATA
1) LATCH DATA
2) NEGATE UDS OR LDS
3) START DATA MODIFICATION
TERMINATE THE CYCLE
1) REMOVE DATA FROM D0-D7 OR D8-D15
2) NEGATE DTACK
START OUTPUT TRANSFER
1) SET R/W TO WRITE
2) PLACE DATA ON D0-D7 OR D8-D15
3) ASSERT UPPER DATA STROBE (UDS) OR LOWER DATA STROBE (LDS)
INPUT THE DATA
1) STORE DATA ON D0-D7 OR D8-D15
2) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
TERMINATE OUTPUT TRANSFER
1) NEGATE UDS OR LDS
2) NEGATE AS
3) REMOVE DATA FROM D0-D7 OR D8-D15
4) SET R/W TO READ
TERMINATE THE CYCLE
1) NEGATE DTACK
START NEXT CYCLE
PF00271S
Bus Master
Slave

Figure 16. Read-Modify-Write Cycle Flowchart

Read-Modify-Write Cycle — The read-modify-write cycle performs a read, modifies the data in the arithmetic-logic unit, and writes the data back to the same address. In the SCN68010, this cycle is indivisible in that the address strobe is asserted throughout the entire cycle. The test and set (TAS) instruction uses this cycle to provide meaningful communication between processors in a multiple processor environment. This instruction is the only instruction that uses the read-modify-write cycle and since the test and set instruction only operates on bytes, all read-modify-write cycles are byte operations. A read-modify-write flowchart is given in Figure 16 and a timing diagram is given in Figure 17.

Wait cycles will be inserted between S4 and S5 on the read portion of the bus cycle and between S16 and S17 on the write portion of the cycle if $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, or $\overline{\text{VPA}}$ is not asserted for the required setup time prior to the falling edge of S4 and S16 respectively.

CPU Space Cycle

During a CPU space cycle, the SCN68010 reads a peripheral device vector number or indicates a breakpoint instruction. If the cycle is to read a vector number it is referred to as an interrupt acknowledge cycle. A CPU space cycle is indicated when the function codes are all high. The address bus then defines what type of CPU space cycle is being executed. The SCN68010 defines two types of CPU space cycles, the interrupt acknowledge cycle, and the breakpoint cycle.

The interrupt acknowledge cycle on an S68000 family compatible processor is defined as a CPU space cycle with the most significant address lines high; on the SCN68010 this means that A4 – A23 will be high. The level of the interrupt being acknowledged is encoded on address lines A1 – A3. An interrupt acknowledge cycle is terminated in the same manner as a normal read cycle. The processor expects a peripheral device to respond to an interrupt acknowledge cycle with a vector number that will be used to transfer control to an interrupt handler routine. See **Interrupts** for further discussion of the interrupt acknowledge cycle.

The breakpoint read cycle is executed by the SCN68010 in response to a breakpoint illegal instruction. A breakpoint cycle on the SCN68010 is defined as a CPU space cycle with all of the address lines low. The processor does not accept or send any data during this cycle. The breakpoint cycle may be terminated by $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, or $\overline{\text{VPA}}$. See **Illegal and Unimplemented Instructions** for further discussion of breakpoints.

Since all members of the S68000 Family do not implement A20 – A31, these lines do not need to be decoded for CPU space functions. Only A16 – A19 are used to distinguish between different CPU space cycle types. The SCN68010 only uses the $0 and $F CPU space types as shown in figure 18; however, all unused encodings of bits A16 – A19 are reserved by Signetics for future extensions of the CPU space functions.

Bus Arbitration

Bus arbitration is a technique used by master-type devices to request, be granted, and acknowledge bus mastership. In its simplest form, it consists of the following:

1. asserting a bus mastership request,
2. receiving a grant that the bus is available at the end of the current cycle, and
3. acknowledging that mastership has been assumed.

Figure 19 is a flowchart showing the detail involved in a request from a single device. Figure 20 is a timing diagram for the same operation. This technique allows processing of bus requests during data transfer cycles.

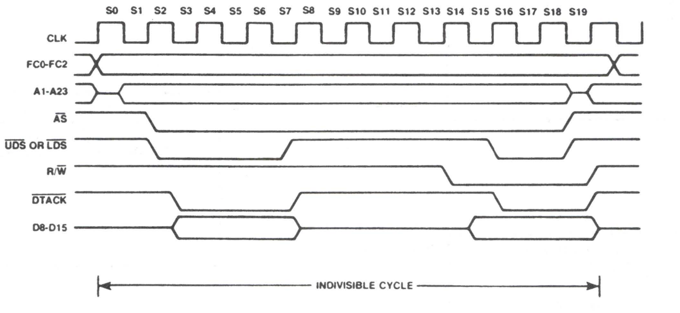

S0 S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14 S15 S16 S17 S18 S19
CLK
FC0-FC2
A1-A23
$\overline{\text{AS}}$
$\overline{\text{UDS}}$ OR $\overline{\text{LDS}}$
R/$\overline{\text{W}}$
$\overline{\text{DTACK}}$
D8-D15
INDIVISIBLE CYCLE
WF01261S

Figure 17. Read-Modify-Write Cycle Timing Diagram

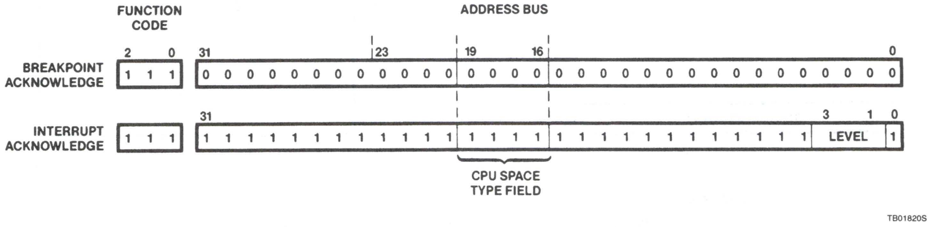

FUNCTION CODE
ADDRESS BUS
2 0
31
23
19
16
0
BREAKPOINT ACKNOWLEDGE
1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
31
3 1 0
INTERRUPT ACKNOWLEDGE
1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LEVEL
1
CPU SPACE TYPE FIELD
TB01820S

Figure 18. SCN68010 CPU-Space Address Encoding

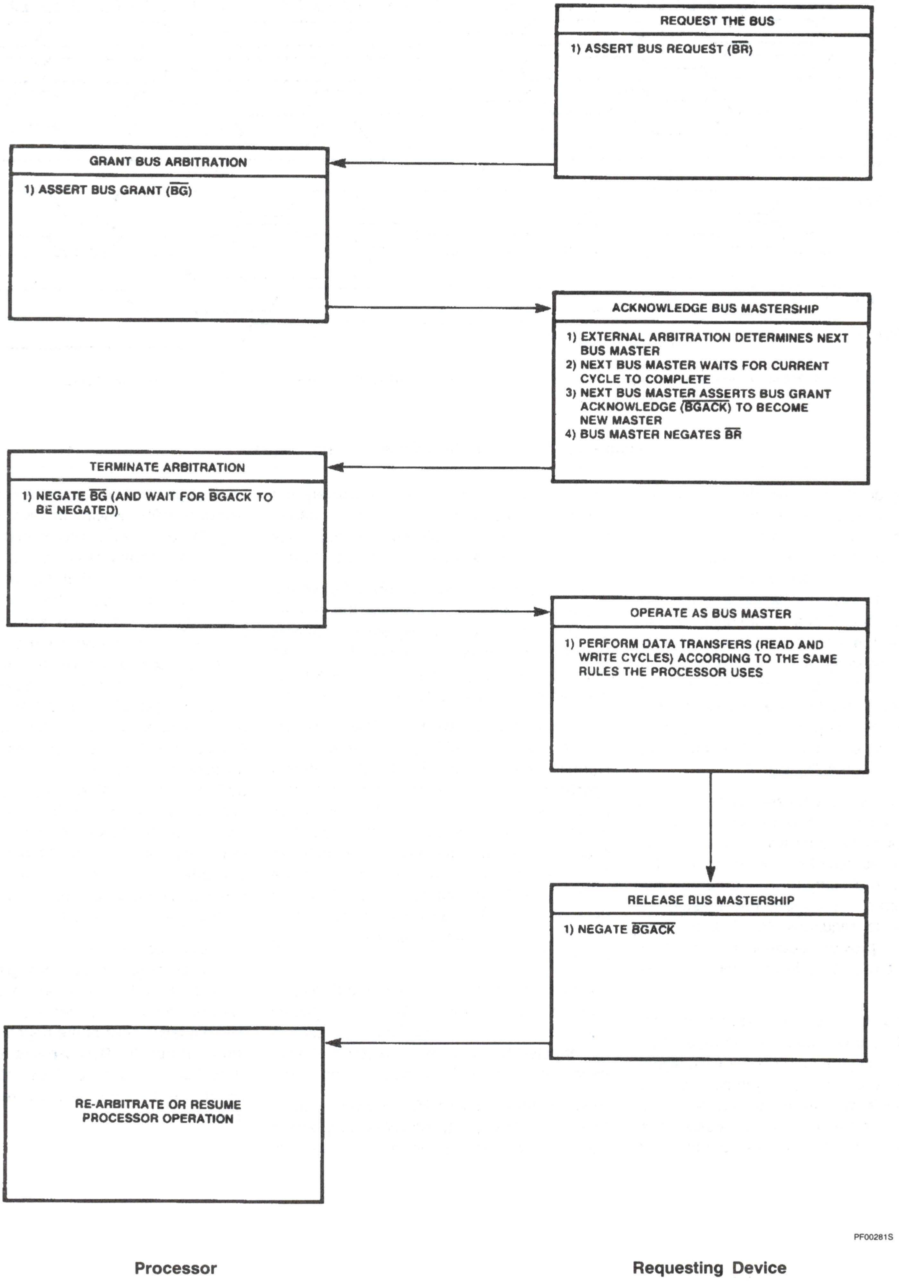

REQUEST THE BUS
1) ASSERT BUS REQUEST ($\overline{BR}$)
GRANT BUS ARBITRATION
1) ASSERT BUS GRANT ($\overline{BG}$)
ACKNOWLEDGE BUS MASTERSHIP
1) EXTERNAL ARBITRATION DETERMINES NEXT BUS MASTER
2) NEXT BUS MASTER WAITS FOR CURRENT CYCLE TO COMPLETE
3) NEXT BUS MASTER ASSERTS BUS GRANT ACKNOWLEDGE ($\overline{BGACK}$) TO BECOME NEW MASTER
4) BUS MASTER NEGATES $\overline{BR}$
TERMINATE ARBITRATION
1) NEGATE $\overline{BG}$ (AND WAIT FOR $\overline{BGACK}$ TO BE NEGATED)
OPERATE AS BUS MASTER
1) PERFORM DATA TRANSFERS (READ AND WRITE CYCLES) ACCORDING TO THE SAME RULES THE PROCESSOR USES
RELEASE BUS MASTERSHIP
1) NEGATE $\overline{BGACK}$
RE-ARBITRATE OR RESUME PROCESSOR OPERATION
PF00281S
Processor
Requesting Device

Figure 19. Bus Arbitration Cycle Flowchart

16-Bit Virtual Memory Microprocessor SCN68010

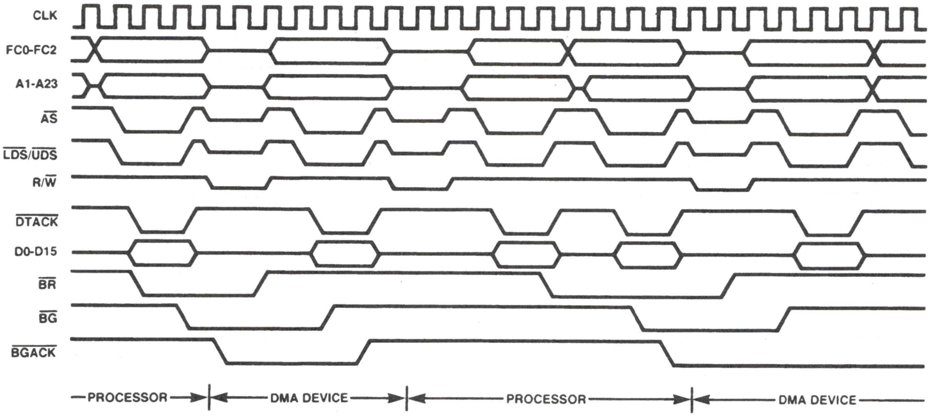


Figure 20. Bus Arbitration Cycle Timing Diagram

The timing diagram shows that the bus request is negated at the time that an acknowledge is asserted. This type of operation would be true for a system consisting of the processor and one device capable of bus mastership. In systems having a number of devices capable of bus mastership, the bus request line from each device is wire ORed to the processor. In this system, it is easy to see that there could be more than one bus request being made. The timing diagram shows that the bus grant signal is negated a few clock cycles after the transition of the acknowledge ($\overline{BGACK}$) signal.

However, if bus requests are still pending, the processor will assert another bus grant within a few clock cycles after it was negated. This additional assertion of bus grant allows external arbitration circuitry to select the next bus master before the current bus master has completed its requirements. The following paragraphs provide additional information about the three steps in the arbitration process.

Requesting the Bus — External devices capable of becoming bus masters request the bus by asserting the bus request ($\overline{BR}$) signal. This is a wire-ORed signal (although it need not be constructed from open-collector devices) that indicates to the processor that some external device requires control of the external bus. The processor is effectively at a lower bus priority level than the external device and will relinquish the bus after it has completed the last bus cycle it has started.

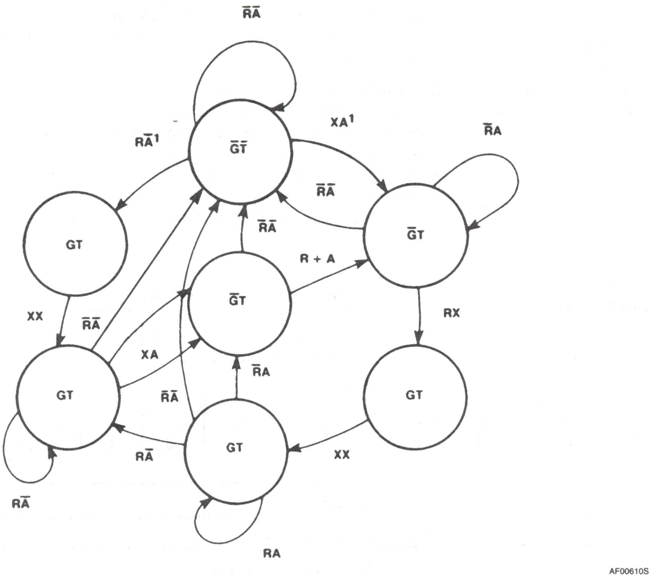
When no acknowledge is received before the bus request signal goes inactive, the processor will continue processing when it detects that the bus request is inactive. This allows ordinary processing to continue if the arbitration circuitry responded to noise inadvertently.

Receiving the Bus Grant — The processor asserts bus grant ($\overline{BG}$) as soon as possible. Normally this is immediately after internal synchronization. The only exception to this occurs when the processor has made an internal decision to execute the next bus cycle but has not progressed far enough into the cycle to have asserted the address strobe ($\overline{AS}$) signal. In this case, bus grant will be delayed until $\overline{AS}$ is asserted to indicate to external devices that a bus cycle is being executed.

The bus grant signal may be routed through a daisy-chained network or through a specific priority-encoded network. The processor is not affected by the external method of arbitration as long as the protocol is obeyed.

Acknowledgement of Mastership — Upon receiving a bus grant, the requesting device waits until address strobe, data transfer acknowledge, and bus grant acknowledge are negated before issuing its own $\overline{BGACK}$. The negation of the $\overline{AS}$ indicates that the previous master has completed its cycle; the negation of $\overline{BGACK}$ indicates that the previous master has released the bus. (While address strobe is asserted, no device is allowed to "break into" a cycle.) The negation of $\overline{DTACK}$ indicates the previous slave has terminated its connection to the previous master. Note that in some applications data transfer acknowledge might not enter into this function. General purpose devices would then be connected such that they were only dependent on address strobe. When bus grant acknowledge is issued, the device is a bus master until it negates bus grant acknowledge. Bus grant acknowledge should not be negated until after the bus cycle(s) is (are) completed. Bus mastership is terminated at the negation of bus grant acknowledge.

The bus request from the granted device should be negated after bus grant acknowledge is asserted. If a bus request is still pending, another bus grant will be asserted within a few clocks of the negation of the bus grant. Refer to **Bus Arbitration Control.** Note that the processor does not perform any external bus cycles before it re-asserts bus grant.


$\overline{R}\,\overline{A}$
$\overline{G}\,\overline{T}$
$R\overline{A}^1$
XA^1
$\overline{R}A$
$\overline{R}\,\overline{A}$
GT
$\overline{G}T$
$\overline{R}\,\overline{A}$
R + A
$\overline{G}T$
XX
$\overline{R}\,\overline{A}$
RX
XA
$\overline{R}A$
GT
$\overline{R}\,\overline{A}$
GT
GT
$R\overline{A}$
XX
$R\overline{A}$
RA
AF00610S

NOTES:
R = Bus request internal
A = Bus grant acknowledge internal
G = Bus grant
T = 3-State control to bus control logic[2]
X = Don't care
1. State machine will not change if the bus is S0 or S1. Refer to Bus Arbitration Control.
2. The address bus will be placed in the high-impedance state if T is asserted and $\overline{AS}$ is negated.

Figure 21. SCN68010 Bus Arbitration Unit State Diagram

Bus Arbitration Control

The bus arbitration control unit in the SCN68010 is implemented with a finite state machine. A state diagram of this machine is shown in Figure 21. All asynchronous signals to the SCN68010 are synchronized before they are used internally. This synchronization is accomplished in a maximum of one cycle of the system clock, assuming that the asynchronous input setup time (#47) has been met (see Figure 22). The input signal is sampled on the falling edge of the clock and is valid internally after the next rising edge.

As shown in Figure 21, input signals labeled R and A are internally synchronized on the bus request and bus grant acknowledge pins respectively. The bus grant output is labeled G and the internal 3-State control signal T. If T is true, the address, data, and control buses are placed in a high-impedance state when $\overline{AS}$ is negated. All signals are shown in positive logic (active high) regardless of their true active voltage level.

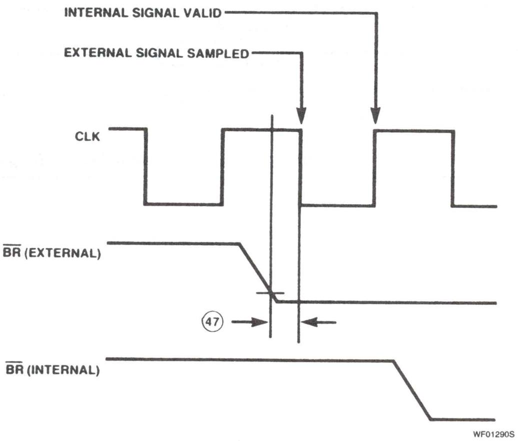

INTERNAL SIGNAL VALID
EXTERNAL SIGNAL SAMPLED
CLK
$\overline{BR}$ (EXTERNAL)
(47)
$\overline{BR}$ (INTERNAL)
WF01290S

Figure 22. Timing Relationship of External Asynchronous Inputs to Internal Signals

State changes (valid outputs) occur on the next rising edge after the internal signal is valid.

A timing diagram of the bus arbitration sequence during a processor bus cycle is shown in Figure 23. The bus arbitration sequence while the bus is inactive (i.e., executing internal operations such as a multiply instruction) is shown in Figure 24.

If a bus request is made at a time when the MPU has already begun a bus cycle but $\overline{AS}$ has not been asserted (bus state S0), $\overline{BG}$ will not be asserted on the next rising edge. Instead, $\overline{BG}$ will be delayed until the second rising edge following its internal assertion. This sequence is shown in Figure 25.

Bus Error and Halt Operation

In a bus architecture that requires a handshake from an external device, the possibility exists that the handshake might not occur. Since different systems will require a different maximum response time, a bus error input is provided. External circuitry must be used to determine the duration between address strobe and data transfer acknowledge before issuing a bus error signal. When a bus error or/and halt signal is received, the processor will initiate a bus error exception sequence or try to rerun the bus cycle.

In addition to a bus timeout indicator, the bus error input is used to indicate a page fault in a virtual memory system. When an external memory management unit detects an invalid access, a bus error is signaled to suspend execution of the current instruction.

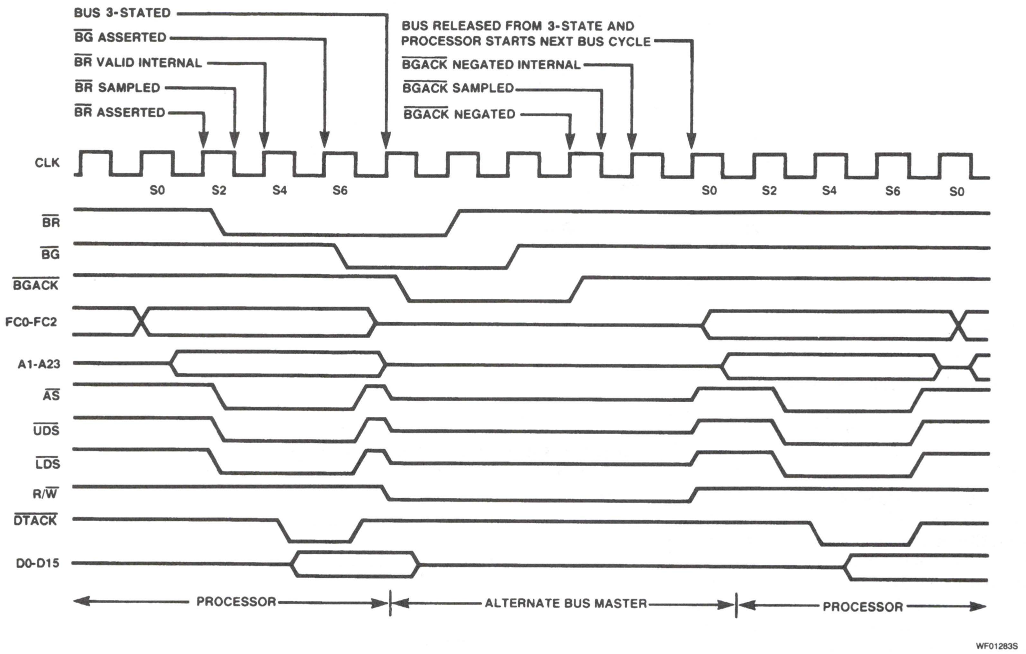

BUS 3-STATED
BG ASSERTED
BR VALID INTERNAL
BR SAMPLED
BR ASSERTED
BUS RELEASED FROM 3-STATE AND PROCESSOR STARTS NEXT BUS CYCLE
BGACK NEGATED INTERNAL
BGACK SAMPLED
BGACK NEGATED
CLK
S0
S2
S4
S6
S0
S2
S4
S6
S0
BR
BG
BGACK
FC0-FC2
A1-A23
AS
UDS
LDS
R/W
DTACK
D0-D15
PROCESSOR
ALTERNATE BUS MASTER
PROCESSOR
WF01283S

Figure 23. Bus Arbitration Timing Diagram — Processor Active

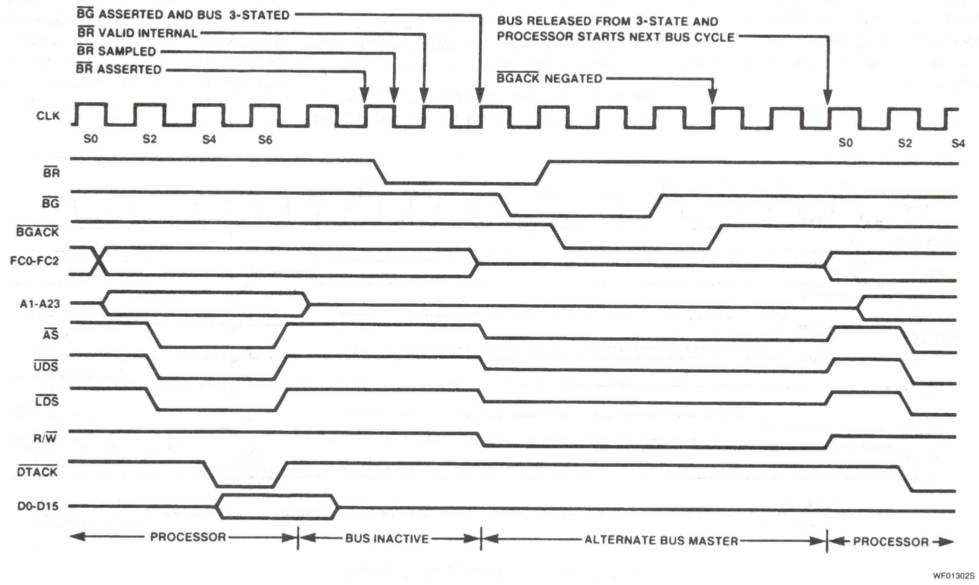

BG ASSERTED AND BUS 3-STATED
BR VALID INTERNAL
BR SAMPLED
BR ASSERTED
BUS RELEASED FROM 3-STATE AND PROCESSOR STARTS NEXT BUS CYCLE
BGACK NEGATED
CLK
S0
S2
S4
S6
S0
S2
S4
BR
BG
BGACK
FC0-FC2
A1-A23
AS
UDS
LDS
R/W
DTACK
D0-D15
PROCESSOR
BUS INACTIVE
ALTERNATE BUS MASTER
PROCESSOR
WF01302S

Figure 24. Bus Arbitration Timing Diagram — Bus Inactive

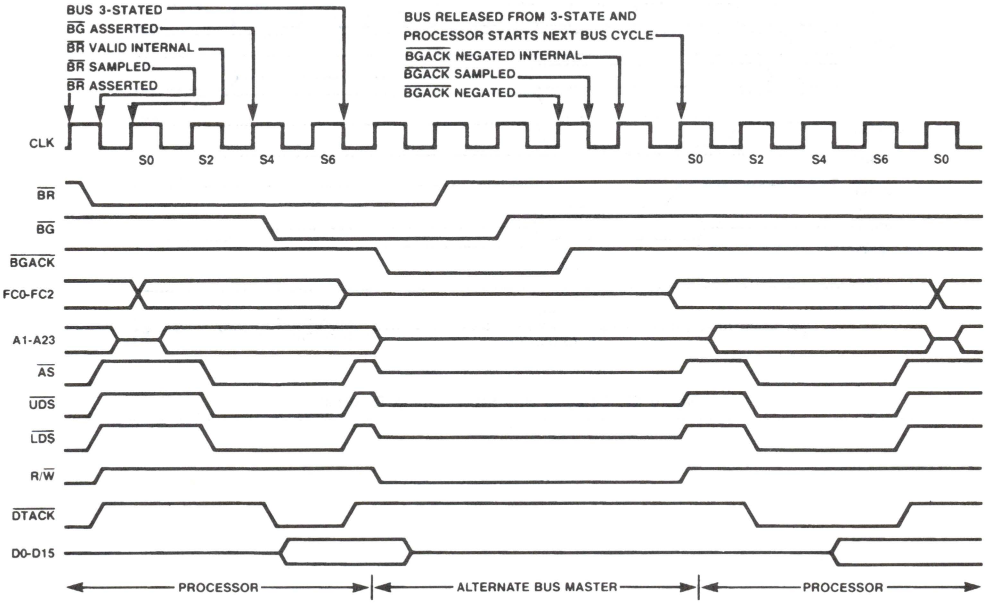

BUS 3-STATED
BG ASSERTED
BR VALID INTERNAL
BR SAMPLED
BR ASSERTED
BUS RELEASED FROM 3-STATE AND PROCESSOR STARTS NEXT BUS CYCLE
BGACK NEGATED INTERNAL
BGACK SAMPLED
BGACK NEGATED
CLK
S0
S2
S4
S6
S0
S2
S4
S6
S0
BR
BG
BGACK
FC0-FC2
A1-A23
AS
UDS
LDS
R/W
DTACK
D0-D15
PROCESSOR
ALTERNATE BUS MASTER
PROCESSOR

Figure 25. Bus Arbitration Timing Diagram — Special Case

16-Bit Virtual Memory Microprocessor SCN68010

Bus Error Operation — When the bus error signal is used to terminate a bus cycle, the SCN68010 will enter exception processing immediately following the bus cycle. The bus error signal is recognized in either of the following cases:

1. $\overline{\text{DTACK}}$ and $\overline{\text{HALT}}$ are negated and $\overline{\text{BERR}}$ is asserted.
2. $\overline{\text{HALT}}$ and $\overline{\text{BERR}}$ are negated and $\overline{\text{DTACK}}$ is asserted. $\overline{\text{BERR}}$ is then asserted within one clock cycle.

When the bus error condition is recognized, the current bus cycle will be terminated in S9 for a read cycle, a write cycle, or the read portion of a read-modify-write cycle and in S21 of the write portion of a read-modify-write cycle. As long as $\overline{\text{BERR}}$ remains asserted, the data and address buses will be in the high-impedance state. Figures 26 and 27 show the timing diagrams for both types of bus error signals.

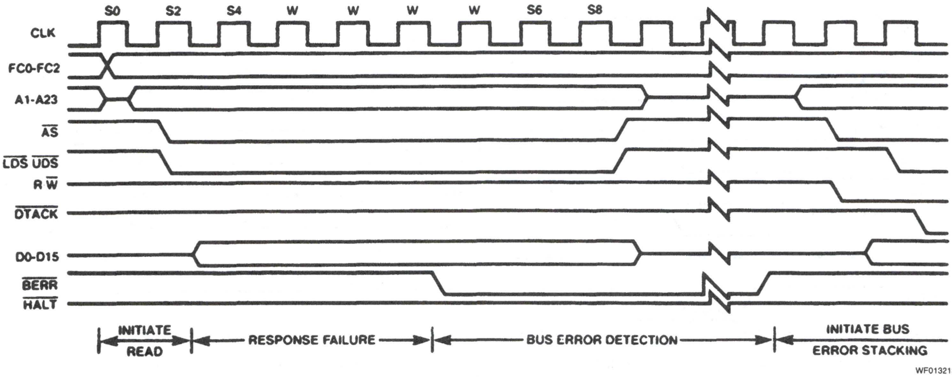

S0
S2
S4
W
W
W
W
S6
S8
CLK
FC0-FC2
A1-A23
AS
LDS UDS
R W
DTACK
D0-D15
BERR
HALT
INITIATE READ
RESPONSE FAILURE
BUS ERROR DETECTION
INITIATE BUS ERROR STACKING
WF01321S

Figure 26. Bus Error Timing Diagram

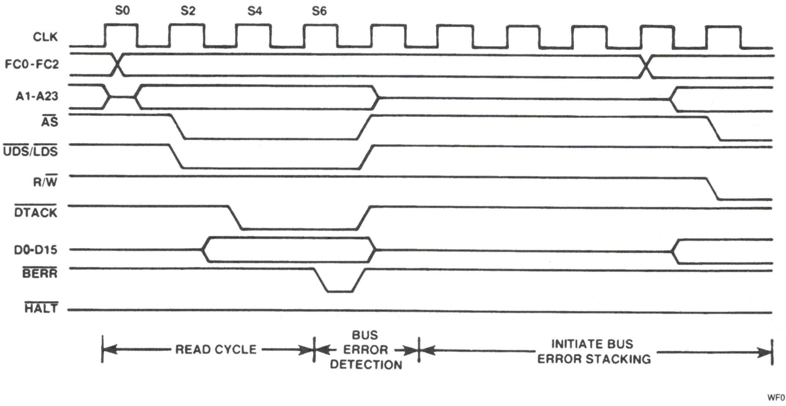

S0
S2
S4
S6
CLK
FC0-FC2
A1-A23
AS
UDS/LDS
R/W
DTACK
D0-D15
BERR
HALT
READ CYCLE
BUS ERROR DETECTION
INITIATE BUS ERROR STACKING
WF01331S

Figure 27. Delayed Bus Error Timing Diagram

After the aborted bus cycle is terminated and $\overline{\text{BERR}}$ is negated, the SCN68010 enters exception processing for the bus error exception. During the exception processing sequence, the following information is placed on the supervisor stack:

1. Status register
2. Program counter (two words, may be up to five words past the instruction being executed)
3. Frame format and vector offset
4. Internal register information, 22 words

Note that the first four words of information are identical to the information stacked by any other exception such as an interrupt or TRAP instruction. The additional information is used by the SCN68010 to continue the execution of the suspended instruction when it is reloaded by an RTE instruction. See **Bus Error** for further details.

After the SCN68010 has placed the above information on the stack, the bus error exception vector is read from vector table entry number two (offset $08) and placed in the program counter. The processor then resumes instruction execution.

NOTE

If a read-modify-write instruction is terminated with a bus error and later continued with an RTE instruction, the processor will rerun the entire cycle whether the bus error occurred on the read or the write portion of the cycle.

Rerun Operation — When, during a bus cycle, the processor receives a bus error signal and the halt pin is being driven by an external device, the processor enters the rerun sequence. A delayed rerun signal may be used similarly to the delayed bus error signal described above. Figures 28 and 29 are timing diagrams for both methods of rerunning the bus cycle.

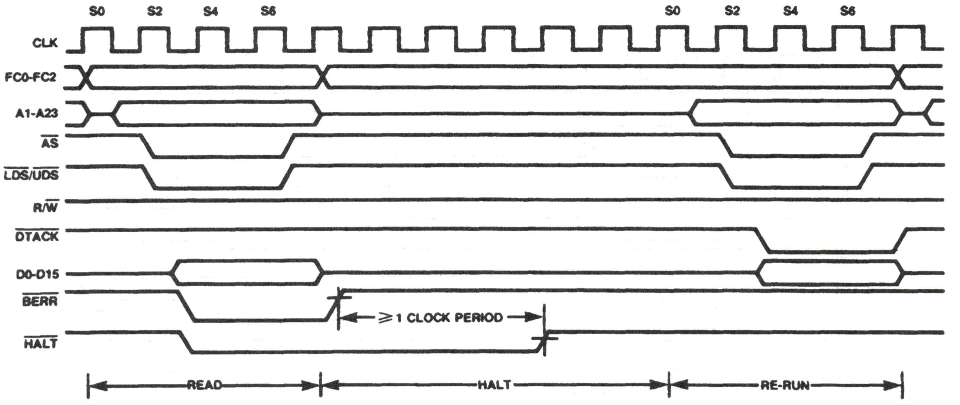


Figure 28. Rerun Bus Cycle Timing Diagram

The processor terminates the bus cycle, then puts the address and data lines in the high-impedance state. The processor remains "halted", and will not run another bus cycle until the halt signal is removed by external logic. Then the processor will rerun the previous cycle using the same function codes, the same data (for a write operation), and the same address. The bus error signal should be removed at least one clock cycle before the halt signal is removed.

NOTE

The processor will not rerun a read-modify-write cycle. This restriction is made to guarantee that the entire cycle runs correctly and that the write operation of a test-and-set operation is performed without ever releasing $\overline{AS}$. If $\overline{BERR}$ and $\overline{HALT}$ are asserted during a read-modify-write bus cycle, a bus error operation results.

Halt Operation — The halt input signal to the SCN68010 performs a halt/run/single-step function in a similar fashion to the synchronous halt function. The halt and run modes are somewhat self explanatory in that when the halt signal is constantly active the processor "halts" (does nothing) and when the halt signal is constantly inactive the processor "runs" (does something).

This single-step mode is derived from correctly timed transitions on the halt signal input. It forces the processor to execute a single bus cycle by entering the run mode until the processor starts a bus cycle then changing to the halt mode. Thus, the single-step mode allows the user to proceed through (and therefore debug) processor operations one bus cycle at a time.

Figure 30 details the timing required for correct single-step operations. Some care must be exercised to avoid harmful interactions between the bus error signal and the halt pin when using the single-cycle mode as a debugging tool. This is also true of interactions between the halt and reset lines since these can reset the machine.

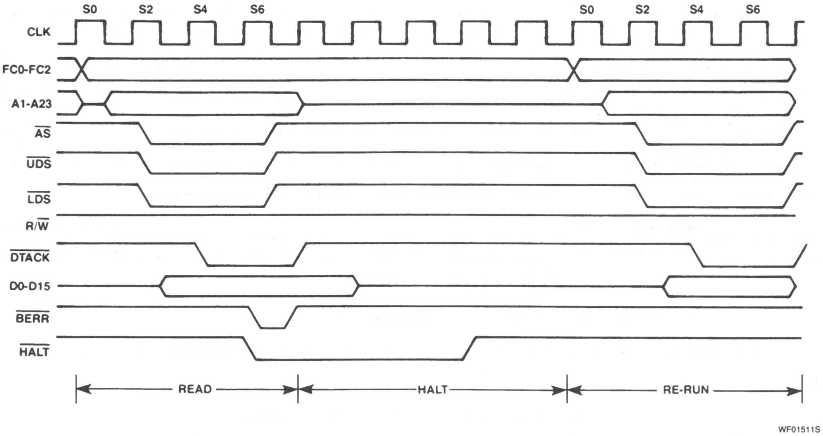


Figure 29. Delayed Rerun Bus Cycle Timing Diagram

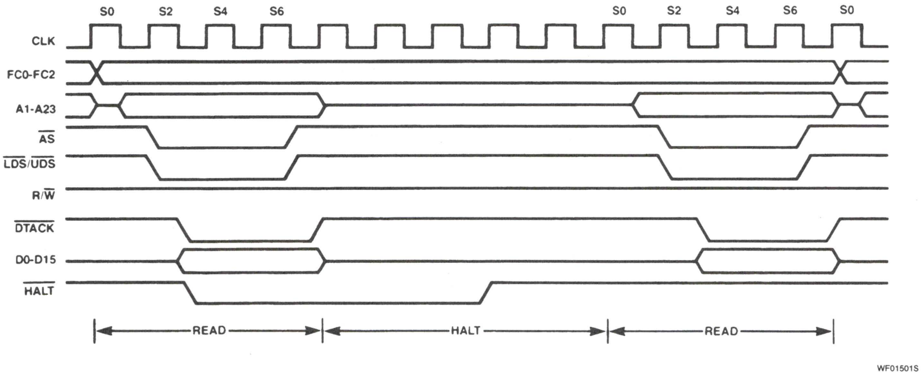


Figure 30. Halt Processor Timing Diagram

When the processor completes a bus cycle after recognizing that the halt signal is active, most 3-State signals are put in the high-impedance state, these include:

1. address lines, and
2. data lines.

This is required for correct performance of the rerun bus cycle operation.

While the processor is honoring the halt request, bus arbitration performs as usual. That is, halting has no effect on bus arbitration. It is the bus arbitration function that removes the control signals from the bus.

The halt function and the hardware trace capability allow the hardware debugger to trace single bus cycles or single instructions at a time. These processor capabilities, along with a software debugging package, give total debugging flexibility.

Double Bus Faults — When a bus error exception occurs, the processor will attempt to stack several words containing information about the state of the machine. If a bus error exception occurs during the stacking operation, there have been two bus errors in a row. This is commonly referred to as a double bus fault. When a double bus fault occurs, the processor will halt and drive the $\overline{\text{HALT}}$ line low. Once a bus error exception has occurred, any bus error exception occurring before the execution of the next instruction constitutes a double bus fault.

Note that a bus cycle which is rerun does not constitute a bus error exception and does not contribute to a double bus fault. Note also that this means that as long as the external hardware requests it, the processor will continue to rerun the same bus cycle.

The bus error pin also has an effect on processor operation after the processor receives an external reset input. The processor reads the vector table after a reset to determine the address to start program execution. If a bus error occurs while reading the vector table (or at any time before the first instruction is executed), the processor reacts as if a double bus fault has occurred and it halts. Only an external reset will start a halted processor.

Reset Operation

The reset signal is a bidirectional signal that allows either the processor or an external device to reset the system. Figure 31 is a timing diagram for the reset operation. Both the halt and reset lines must be asserted to ensure total reset of the processor in all cases.

When the reset and halt lines are driven by an external device, it is recognized as an entire system reset, including the processor. The processor responds by reading the reset vector table entry (vector number zero, address $000000) and loads it into the supervisor stack pointer (SSP). Vector table entry number one at address $000004 is read next and loaded into the program counter. The processor initializes the status register to an interrupt level of seven and the vector base register to $00000000. No other registers are affected by the reset sequence.

When a reset instruction is executed, the processor drives the reset pin for 124 clock periods. In this case, the processor is trying to reset the rest of the system. Therefore, there is no effect on the internal state of the processor. All of the processor's internal registers and the status register are unaffected by the execution of a reset instruction. All external devices connected to the reset line should be reset at the completion of the reset instruction.

Asserting the $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ lines for ten clock cycles will cause a processor reset, except when V_{CC} is initially applied to the processor. In this case, an external reset must be applied for at least 100ms.

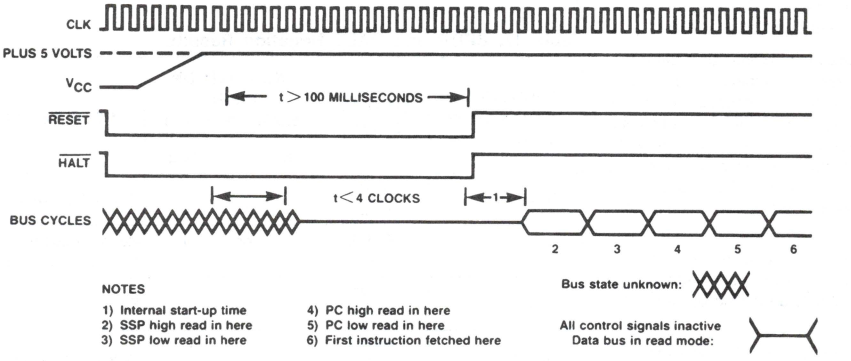


Figure 31. Reset Operation Timing Diagram

The Relationship of DTACK, BERR, and HALT

In order to properly control termination of a bus cycle for a rerun or a bus error condition, DTACK, BERR, and HALT should be asserted and negated on the rising edge of the SCN68010 clock. This will assure that when two signals are asserted simultaneously, the required setup time (#47) for both of them will be met during the same bus state. This, or some equivalent precaution, should be designed external to the SCN68010.

The preferred bus cycle terminations may be summarized as follows (case numbers refer to Table 16):

Normal Termination:
DTACK is asserted, BERR and HALT remain negated (case 1).

Halt Termination:
HALT is asserted at same time, or before DTACK and BERR remains negated (case 2).

Bus Error Termination:
BERR is asserted in lieu of, at the same time, or before DTACK (case 3) or after DTACK (case 4) and HALT remains negated; BERR is negated at the same time or after DTACK.

Rerun Termination:
HALT and BERR are asserted in lieu of, at the same time, or before DTACK (case 5) or after DTACK (case 6); BERR is negated at the same time or after DTACK, HALT must be held at least one cycle after BERR.

Table 16 details the resulting bus cycle termination under various combinations of control signal sequences. The negation of these same control signals under several conditions is shown in Table 17. (DTACK is assumed to be negated normally in all cases; for best results, both DTACK and BERR should be negated when address strobe is negated).

EXAMPLE A:
A system uses a watch-dog timer to terminate accesses to unpopulated address space. The timer asserts BERR after time out (case 3).

EXAMPLE B:
A system uses error detection on RAM contents. Designer may:

a) Delay DTACK until data verified, and return BERR and HALT simultaneously to rerun error cycle (case 5), or if valid, return DTACK (case 1).

b) Delay DTACK until data verified, and return BERR at same time as DTACK if data in error (case 3).

c) Return DTACK prior to data verification, as described in the next section. If data is invalid, BERR is asserted on next clock cycle (case 4).

d) Return DTACK prior to data verification, if data is invalid assert BERR and HALT on next clock cycle (case 6). The memory controller may then correct the RAM prior to or during the rerun.

Table 16. DTACK, BERR, and HALT Assertion Results

CASE NO.	CONTROL SIGNAL	ASSERTED ON RISING EDGE OF STATE		RESULT
		N	N+2	
1	DTACK BERR HALT	A NA NA	S NA X	Normal cycle terminate and continue.
2	DTACK BERR HALT	A NA A/S	S NA S	Normal cycle terminate and halt. Continue when HALT removed.
3	DTACK BERR HALT	X A NA	X S NA	Terminate and take bus error trap.
4	DTACK BERR HALT	A NA NA	X A NA	Terminate and take bus error trap.
5	DTACK BERR HALT	X A A/S	X S S	Terminate and rerun when HALT removed.
6	DTACK BERR HALT	A NA NA	X A A	Terminate and rerun when HALT removed.

NOTES:
N – the number of the current even bus state (e.g., S4, S6, etc.)
A – signal is asserted in this bus state
NA – signal is not asserted in this state
X – don't care
S – signal was asserted in previous state and remains asserted in this state

Table 17. BERR and HALT Negation Results

CONDITIONS OF TERMINATION IN TABLE 16	CONTROL SIGNAL	NEGATED ON RISING EDGE OF STATE			RESULTS – NEXT CYCLE
		N		N+2	
Bus Error	BERR	•	or	•	Takes bus error trap.
	HALT	•	or	•	
Rerun	BERR	•	or	•	Illegal sequence; usually traps to vector number 0.
	HALT	•			
Rerun	BERR	•			Reruns the bus cycle.
	HALT			•	
Normal	BERR	•			May lengthen next cycle.
	HALT	•	or	•	

NOTE:
• = Signal is negated in this bus state.

Asynchronous Versus Synchronous Operation

Asynchronous Operation

To achieve clock frequency independence at a system level, the SCN68010 can be used in an asynchronous manner. This entails using only the bus handshake lines (AS, UDS, LDS, DTACK, BERR, HALT and VPA) to control the data transfer. Using this method, AS signals the start of a bus cycle and the data strobes are used as a condition for valid data on a write cycle. The slave device (memory or peripheral) then responds by placing the requested data on the data bus for a read cycle or latching data on a write cycle and asserting the data transfer acknowledge signal (DTACK) to terminate the bus cycle. If no slave responds or the access is invalid, external control logic asserts the BERR, or BERR and HALT, signal to abort or rerun the bus cycle.

The DTACK signal is allowed to be asserted before the data from a slave device is valid on a read cycle. The length of time that DTACK may precede data is given as parameter #31 (See **AC Electrical Characteristics** for # references) and it must be met in any asynchronous system to insure that valid data is latched into the processor. Notice that there is no maximum time specified from the assertion of AS to the assertion of DTACK. This is because the MPU will insert wait cycles of one clock period each until DTACK is recognized.

The BERR signal is allowed to be asserted after the DTACK signal is asserted. BERR must be asserted within the time given as parameter #48 after DTACK is asserted in any asynchronous system to insure proper operation. If this maximum delay time is violated, the processor may exhibit erratic behavior.

Synchronous Operation

To allow for those systems which use the system clock as a signal to generate DTACK and other asynchronous inputs, the asynchronous input setup time is given as parameter #47. If this setup is met on an input, such as DTACK, the processor is guaranteed to recognize that signal on the next falling edge of the system clock. However, the converse is not true — if the input signal does not meet the setup time it is not guaranteed not to be recognized. In addition, if DTACK is recognized on a falling edge, valid data will be latched into the processor (on a read cycle) on the next falling edge provided that the data meets the setup time given as parameter #27. Given this, parameter #31 may be ignored. Note that if DTACK is asserted, with the required setup time, before the falling edge of S4, no wait states will be incurred and the bus cycle will run at its maximum speed of four clock periods.

In order to assure proper operation in a synchronous system when BERR is asserted after DTACK, BERR must meet the setup time parameter #27A prior to the falling edge of the clock one clock cycle after DTACK was recognized. This setup time is critical to proper operation, and the SCN68010 may exhibit erratic behavior if it is violated.

NOTE

During an active bus cycle, VPA and BERR are sampled on every falling edge of the clock starting with S0. DTACK is sampled on every falling edge of the clock starting with S4 and data is latched on the falling edge of S6 during a read. The bus cycle will then be terminated in S7 except when BERR is asserted in the absence of DTACK, in which case it will terminate one clock cycle later in S9.

PROCESSING STATES

This section describes the actions of the SCN68010 which are outside the normal processing associated with the execution of instructions. The functions of the bits in the supervisor portion of the status register are covered: the supervisor/user bit, the trace enable bit, and the processor interrupt priority mask. Finally, the sequence of memory references and actions taken by the processor on exception conditions are detailed.

The SCN68010 is always in one of three processing states: normal, exception, or halted. The normal processing state is that associated with instruction execution; the memory references are to fetch instructions and operands, and to store results. Two special cases of the normal state are the stopped state, which the processor enters when a STOP instruction is executed, and the loop mode, which the processor may enter when a DBcc instruction is executed. In the stopped state, no further memory references are made and in the loop mode only operand references are made.

The exception processing state is associated with interrupts, trap instructions, tracing and other exceptional conditions. The exception may be internally generated by an instruction or by an unusual condition arising during the execution of an instruction. Externally, exception processing can be forced by an interrupt, by a bus error, or by a reset. Exception processing is designed to provide an efficient context switch so that the processor may handle unusual conditions.

The halted processing state is an indication of catastrophic hardware failure. For example, if during the exception processing of a bus error another bus error occurs, the processor assumes that the system is unusable and halts. Only an external reset can restart a halted processor. Note that a processor in the stopped state is not in the halted state, nor vice versa.

Privilege States

The processor operates in one of two states of privilege: the "supervisor" state or the "user" state. The privilege state determines which operations are legal, are used to choose between the supervisor stack pointer and the user stack pointer in instruction references, and may be used by an external memory management device to control and translate accesses.

The privilege state is a mechanism for providing security in a computer system. Programs should access only their own code and data areas, and ought to be restricted from accessing information which they do not need and must not modify.

The privilege mechanism provides security by allowing most programs to execute in user state. In this state, the accesses are controlled, and the effects on other parts of the system are limited. The operating system executes in the supervisor state, has access to all resources, and performs the overhead tasks for the user programs.

Supervisor State

The supervisor state is the higher state of privilege. For instruction execution, the supervisor state is determined by the S bit of the status register; if the S bit is asserted (high), the processor is in the supervisor state. All instructions can be executed in the supervisor state. The bus cycles generated by instructions executed in the supervisor state are classified as supervisor references. While the processor is in the supervisor privilege state, those instructions which use either the system stack pointer implicitly or address register seven explicitly access the supervisor stack pointer.

All exception processing is done in the supervisor state, regardless of the previous setting of the S bit. The bus cycles generated during exception processing are classified as supervisor references. All stacking operations during exception processing use the supervisor stack pointer.

User State

The user state is the lower state of privilege. For instruction execution, the user state is determined by the S bit of the status register; if the S bit is negated (low), the processor is executing instructions in the user state.

Most instructions execute the same in user state as in the supervisor state. However, some instructions which have important system effects are made privileged. User programs are not permitted to execute the STOP instruction, or the RESET instruction. To ensure that a user program cannot enter the supervisor state except in a controlled manner, the instructions which modify the whole status register are privileged. To aid in debugging programs which are to be used as operating systems, the move from status register (MOVE from SR), move to/from user stack pointer (MOVE USP), move to/from control register (MOVEC), and move alternate address space (MOVES) instructions are also privileged.

The bus cycles generated by an instruction executed in the user state are classified as user state references. This allows an external memory management device to translate the address and to control access to protected portions of the address space. While the

16-Bit Virtual Memory Microprocessor SCN68010

Table 18. Bus Cycle Classification

FUNCTION CODE OUTPUT			REFERENCE CLASS
FC2	FC1	FC0	
0	0	0	Unassigned, reserved*
0	0	1	User data
0	1	0	User program
0	1	1	Unassigned, reserved*
1	0	0	Unassigned, reserved*
1	0	1	Supervisor data
1	1	0	Supervisor program
1	1	1	CPU space

NOTE:
*Address space 3 is reserved for user definition, while 0 and 4 are reserved for future use by Signetics.

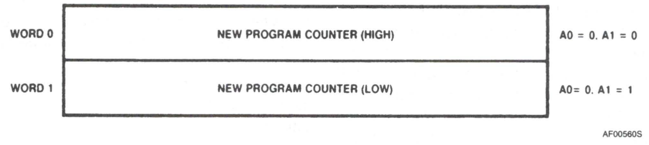
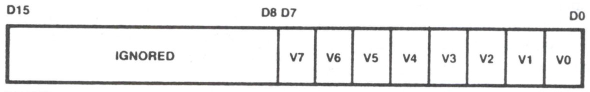


Figure 32. Format of Vector Table Entries



WHERE:
V7 IS THE MSB OF THE VECTOR NUMBER
V0 IS THE LSB OF THE VECTOR NUMBER

AF00550S

Figure 33. Vector Number Format

processor is in the user privilege state, those instructions which use either the system stack pointer implicitly or address register seven explicitly, access the user stack pointer.

Privilege State Changes

Once the processor is in the user state and executing instructions, only exception processing can change the privilege state. During exception processing, the previous setting of the S bit of the status register is saved and the S bit is asserted, putting the processor in the supervisor state. Therefore, when instruction execution resumes at the address specified to process the exception, the processor is in the supervisor privilege state.

Reference Classification

When the processor makes a reference, it classifies the kind of reference being made, using the encoding on the three function code output lines. This allows external translation of addresses, control of access, and differentiation of special processor state, such as interrupt acknowledge. Table 18 lists the classification of references.

Exception Processing

Before discussing the details of interrupts, traps, and tracing, a general description of exception processing is in order. The processing of an exception occurs in four steps, with variations for different exception causes. During the first step, a temporary copy of the status register is made and the status register is set for exception processing. In the second step the exception vector is determined and the third step is the saving of the current processor context. In the fourth step a new context is obtained and the processor resumes instruction processing.

Exception Vectors

Exception vectors are memory locations from which the processor fetches the address of a routine which will handle that exception. All exception vectors are two words in length (Figure 32), except for the reset vector, which is four words. All exception vectors lie in the supervisor data space, except for the reset vector which is in the supervisor program space. A vector number is an 8-bit number which, when multiplied by four, gives the offset of an exception vector. Vector numbers are generated internally or externally, depending on the cause of the exception. In the case of interrupts, during the interrupt acknowledge bus cycle, a peripheral provides an 8-bit vector number (Figure 33) to the processor on data bus lines D0 through D7. The processor translates the vector number into a full 32-bit offset which is added to the contents of the vector base register to generate the address used to fetch the vector, as shown in Figure 34. The memory layout for exception vectors is given in Table 19.

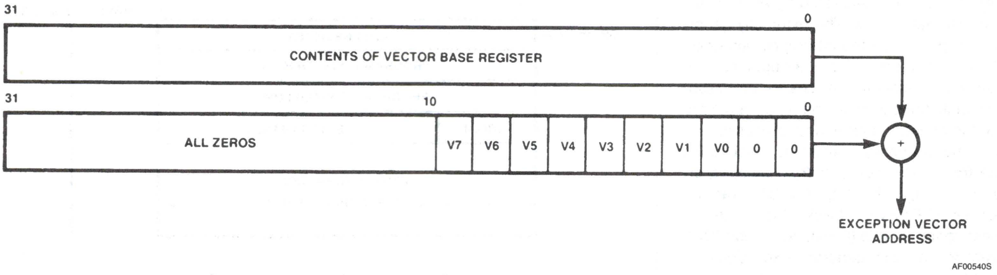


Figure 34. Exception Vector Address Calculation

As shown in Table 19, the memory layout is 512 words long (1024 bytes). It starts at offset 0 and proceeds through offset 1023. This provides 255 unique vectors; some of these are reserved for TRAPs and other system functions. Of the 255, there are 192 reserved for user interrupt vectors. However, there is no protection on the first 63 entries, so externally generated interrupt vector numbers may reference any of the exception vectors at the discretion of the system designer.

Exception Stack Frame

Exception processing saves the most volatile portion of the current processor context on the top of the supervisor stack. This context is organized in a format called the exception stack frame. This information always includes the status register and program counter of the processor when the exception occurred. In order to support generic handlers, the processor also places the vector offset in the exception stack frame. The format code field allows the RTE (return from exception) instruction to identify what information is on the stack so that it may be properly restored. The general form of the exception stack frame is illustrated in Figure 35. Table 20 lists the SCN68010 stack frame codes. Although some formats are peculiar to a particular S68000 family processor, the format 0000 is always legal, and indicates that just the first four words of the frame are present.

Table 20. SCN68010 Format Codes

FORMAT CODE	STACKED INFORMATION
0000	SCN68010 short format (4 words)
1000	SCN68010 long format (29 words)
All others	Unassigned, reserved

Table 19. Exception Vector Table

VECTOR NUMBER(S)	OFFSET			ASSIGNMENT
	DEC	HEX	SPACE	
0	0	000	SP	Reset: initial SSP
—	4	004	SP	Reset: initial PC
2	8	008	SD	Bus error
3	12	00C	SD	Address error
4	16	010	SD	Illegal instruction
5	20	014	SD	Zero divide
6	24	018	SD	CHK instruction
7	28	01C	SD	TRAPV instruction
8	32	020	SD	Privilege violation
9	36	024	SD	Trace
10	40	028	SD	Line 1010 emulator
11	44	02C	SD	Line 1111 emulator
12*	48	030	SD	(Unassigned, reserved)
13*	52	034	SD	(Unassigned, reserved)
14	56	038	SD	Format error
15	60	03C	SD	Uninitialized interrupt vector
16 – 23*	64	04C	SD	(Unassigned, reserved)
	95	05F		—
24	96	060	SD	Spurious interrupt
25	100	064	SD	Level 1 interrupt autovector
26	104	068	SD	Level 2 interrupt autovector
27	108	06C	SD	Level 3 interrupt autovector
28	112	070	SD	Level 4 interrupt autovector
29	116	074	SD	Level 5 interrupt autovector
30	120	078	SD	Level 6 interrupt autovector
31	124	07C	SD	Level 7 interrupt autovector
32 – 47	128	080	SD	TRAP instruction vectors
	191	0BF		—
48 – 63*	192	0C0	SD	(Unassigned, reserved)
	255	0FF		—
64 – 255	256	100	SD	User interrupt vectors

NOTE:

*Vector numbers 12, 13, 16 through 23, and 48 through 63 are reserved for future enhancements by Signetics. No user peripheral devices should be assigned these numbers.

Kinds of Exceptions

Exceptions can be generated by either internal or external causes. The externally generated exceptions are the interrupts, bus error, and reset requests. The interrupts are requests from peripheral devices for processor action while the bus error and reset inputs are used for access control and processor restart. The internally generated exceptions come from instructions, or from address errors or tracing. The trap (TRAP), trap on overflow (TRAPV), check data register against upper bounds (CHK), and divide (DIV) instructions all can generate exceptions as part of their instruction execution. In addition, illegal instructions, word or long word fetches from odd addresses, and privilege violations cause exceptions. Tracing behaves like a very high-priority internally-generated interrupt after each instruction execution.

Exception Processing Sequence

Exception processing occurs in four identifiable steps. In the first step, an internal copy is made of the status register. After the copy is made, the S bit is asserted, putting the processor into the supervisor privilege state. Also, the T bit is negated which will allow the exception handler to execute unhindered by tracing. For the reset and interrupt exceptions, the interrupt priority mask is also updated.

In the second step, the vector number of the exception is determined. For interrupts, the vector number is obtained by a processor fetch classified as an interrupt acknowledge. For all other exceptions, internal logic provides the vector number. This vector number is then used to generate the offset of the exception vector and is added to the vector base register.

The third step is to save the current processor status, except for the reset exception. The exception stack frame is created at the top of the supervisor stack. The current program counter value, the saved copy of the status register, and the format/offset word are written into the stack frame. The program counter value stacked usually points to the next unexecuted instruction; however, for bus error and address error, the value stacked for the program counter is unpredictable, and may be incremented by up to five words from the address of the instruction which caused the error. Group 1 and 2 exceptions (see **Multiple Exceptions**) use a short format exception stack frame (format = 0000). Additional information defining the current context is stacked for the bus error and address error exceptions.

The last step is the same for all exceptions. The new program counter value is fetched from the exception vector table. The processor then resumes instruction execution. The instruction at the address given in the exception vector is fetched, and normal instruction decoding and execution is started.

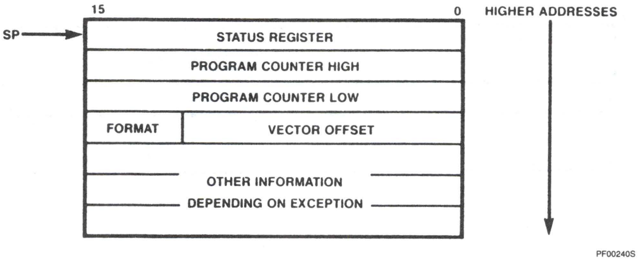


Figure 35. SCN68010 Stack Format

Multiple Exceptions

These paragraphs describe the processing which occurs when multiple exceptions arise simultaneously. Exceptions can be grouped according to their occurrence and priority. The group 0 exceptions are reset, bus error, and address error. These exceptions cause the instruction currently being executed to be aborted and the exception processing to commence within two clock cycles. The group 1 exceptions are trace and interrupt, as well as the privilege violations and illegal instructions. These exceptions allow the current instruction to execute to completion, but pre-empt the execution of the next instruction by forcing exception processing to occur (privilege violations and illegal instructions are detected when they are the next instruction to be executed). The group 2 exceptions occur as part of the normal processing of instructions. The TRAP, TRAPV, CHK, and zero divide exceptions are in this group. For these exceptions, the normal execution of an instruction may lead to exception processing.

Group 0 exceptions have highest priority, while group 2 exceptions have lowest priority. Within group 0, reset has highest priority, followed by address error and then bus error. Within group 1, trace has priority over external interrupts, which in turn takes priority over illegal instruction and privilege violation. Since only one instruction can be executed at a time, there is no priority relation within group 2.

The priority relation between two exceptions determines which is taken, or taken first, if the conditions for both arise simultaneously. Therefore, if a bus error occurs during a TRAP instruction, the bus error takes precedence, and the TRAP instruction processing is suspended. In another example, if an interrupt request occurs during the execution of an instruction while the T bit is asserted, the trace exception has priority, and is processed first. Before instruction processing resumes, however, the interrupt exception is also processed, and instruction processing commences finally in the interrupt handler routine. A summary of exception grouping and priority is given in Table 21.

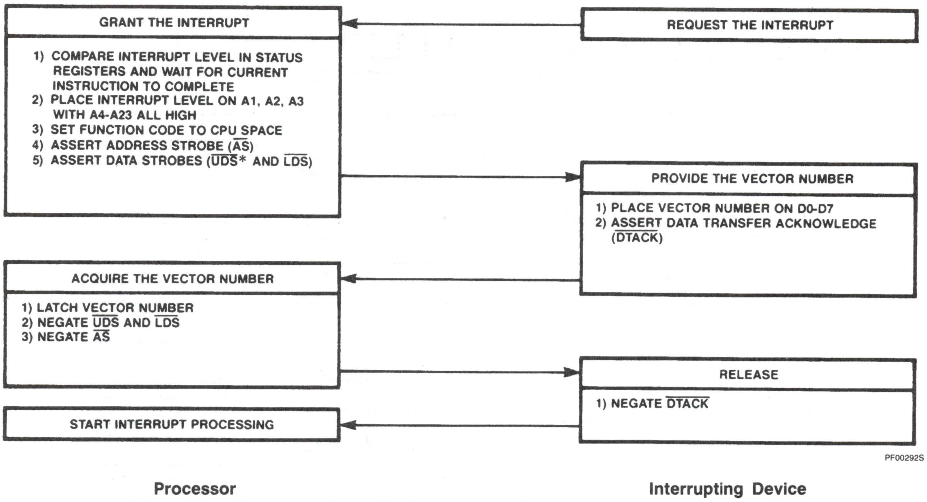
Exception Processing in Detail

Exceptions have a number of sources and each exception has processing which is peculiar to it. The following paragraphs detail the sources of exceptions, how each arises, and how each is processed.

Reset

The reset input provides the highest exception level. The processing of the reset signal is designed for system initiation, and recovery from catastrophic failure. Any processing in progress at the time of the reset is aborted and cannot be recovered. The processor is forced into the supervisor state, the trace state is forced off, and the processor interrupt priority mask is set to level seven. The vector base register is set to $00000000 and the vector number is internally generated to reference the reset exception vector at location 0 in the supervisor program space. Because no assumptions can be made about the validity of register contents , in particular the supervisor stack pointer, neither the program counter nor the status register is saved. The address contained in the first two words of the reset exception vector is fetched as the initial supervisor stack pointer, and the address in the last two words of the reset exception vector is fetched as the initial program counter. Finally, instruction execution is started at the address in the program counter. The power-up/restart code should be pointed to by the initial program counter.

The reset instruction does not cause loading of the reset vector, but does assert the reset line to reset external devices. This allows the software to reset the system to a known state and then continue processing with the next instruction.


GRANT THE INTERRUPT
1) COMPARE INTERRUPT LEVEL IN STATUS REGISTERS AND WAIT FOR CURRENT INSTRUCTION TO COMPLETE
2) PLACE INTERRUPT LEVEL ON A1, A2, A3 WITH A4-A23 ALL HIGH
3) SET FUNCTION CODE TO CPU SPACE
4) ASSERT ADDRESS STROBE (AS)
5) ASSERT DATA STROBES (UDS* AND LDS)
REQUEST THE INTERRUPT
PROVIDE THE VECTOR NUMBER
1) PLACE VECTOR NUMBER ON D0-D7
2) ASSERT DATA TRANSFER ACKNOWLEDGE (DTACK)
ACQUIRE THE VECTOR NUMBER
1) LATCH VECTOR NUMBER
2) NEGATE UDS AND LDS
3) NEGATE AS
RELEASE
1) NEGATE DTACK
START INTERRUPT PROCESSING
PF00292S
Processor
Interrupting Device

NOTE
*Although a vector number is one byte, both data strobes are asserted due to the microcode used for exception processing. The processor does not recognize anything on data lines D8 through D15 at this time.

Figure 36. Vector Acquisition Flowchart

Table 21. Exception Grouping and Priority

GROUP	EXCEPTION	PROCESSING
0	Reset address error bus error	Exception processing begins within two clock cycles
1	Trace interrupt illegal privilege	Exception processing begins before the next instruction
2	TRAP, TRAPV, CHK, zero divide format error	Exception processing is started by normal instruction execution

Interrupts

Seven levels of interrupt priorities are provided. Devices may be chained externally within interrupt priority levels, allowing an unlimited number of peripheral devices to interrupt the processor. Interrupt priority levels are numbered from one to seven, level seven being the highest priority. The status register contains a 3-bit mask which indicates the current processor priority, and interrupts are inhibited for all priority levels less than or equal to the current processor priority.

An interrupt request is made to the processor by encoding the interrupt request level on the interrupt request lines; a zero indicates no interrupt request. Interrupt requests arriving at the processor do not force immediate exception processing, but are made pending. Pending interrupts may cause exception processing to start at the end of an instruction depending on the current processor priority level. If the priority of the pending interrupt is lower than or equal to the current processor priority, execution continues with the next instruction and the interrupt exception processing is postponed. (The recognition of level seven is slightly different, as explained in the following paragraph.)

If the priority of the pending interrupt is greater than the current processor priority, the exception processing sequence is started. A copy of the status register is saved, the privilege state is set to supervisor state, tracing is suppressed, and the processor priority level is set to the level of the interrupt being acknowledged. The processor fetches the vector number from the interrupting device, classifying the reference as an interrupt acknowledge and displaying the level number of the interrupt being acknowledged on the address bus. If external logic requests automatic vectoring, the processor internally generates a vector number which is determined by the interrupt level number. If external logic indicates a bus error, the interrupt is taken to be spurious, and the generated vector number references the spurious interrupt vector. The processor then proceeds with the usual exception processing, saving the format/offset word, program counter, and status register on the supervisor stack. The offset value in the format/offset word is the externally supplied or internally generated vector number multiplied by four. The format will be all zeros. The saved value of the program counter is the address of the instruction which would have been executed had the interrupt not been present. The content of the interrupt vector whose vector number was previously obtained is fetched and loaded into the program counter, and normal instruction execution commences in the interrupt handling routine. A flowchart for the interrupt acknowledge sequence is given in Figure 36, a timing diagram is given in Figure 37, and the interrupt processing sequence is shown in Figure 38.

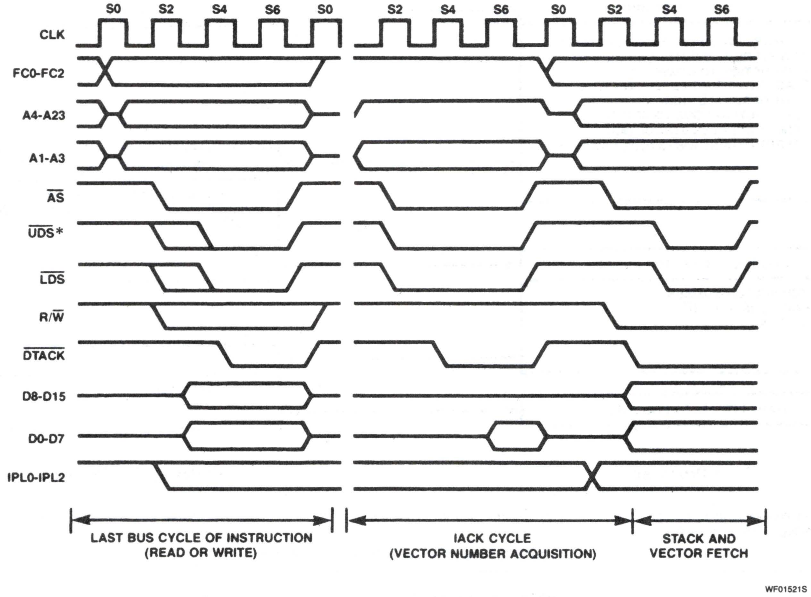

S0
S2
S4
S6
S0
S2
S4
S6
S0
S2
S4
S6
CLK
FC0-FC2
A4-A23
A1-A3
AS
UDS*
LDS
R/W
DTACK
D8-D15
D0-D7
IPL0-IPL2
LAST BUS CYCLE OF INSTRUCTION (READ OR WRITE)
IACK CYCLE (VECTOR NUMBER ACQUISITION)
STACK AND VECTOR FETCH
WF01521S

Figure 37. Interrupt Acknowledge Cycle Timing Diagram

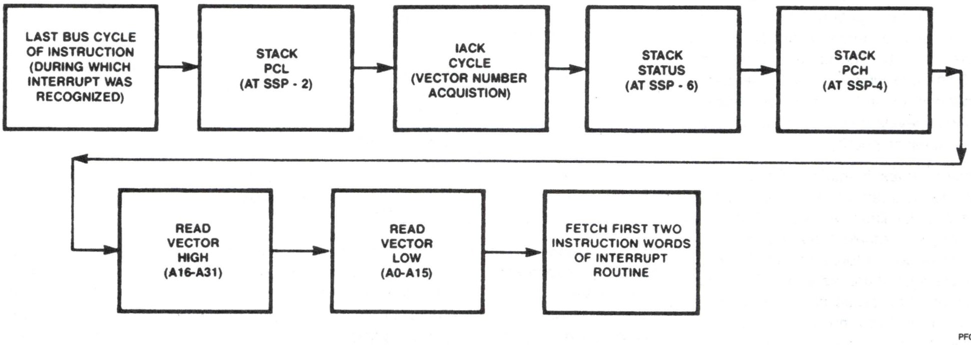

LAST BUS CYCLE OF INSTRUCTION (DURING WHICH INTERRUPT WAS RECOGNIZED)
STACK PCL (AT SSP - 2)
IACK CYCLE (VECTOR NUMBER ACQUISTION)
STACK STATUS (AT SSP - 6)
STACK PCH (AT SSP-4)
READ VECTOR HIGH (A16-A31)
READ VECTOR LOW (A0-A15)
FETCH FIRST TWO INSTRUCTION WORDS OF INTERRUPT ROUTINE
PF00230S

Figure 38. Interrupt Processing Sequence

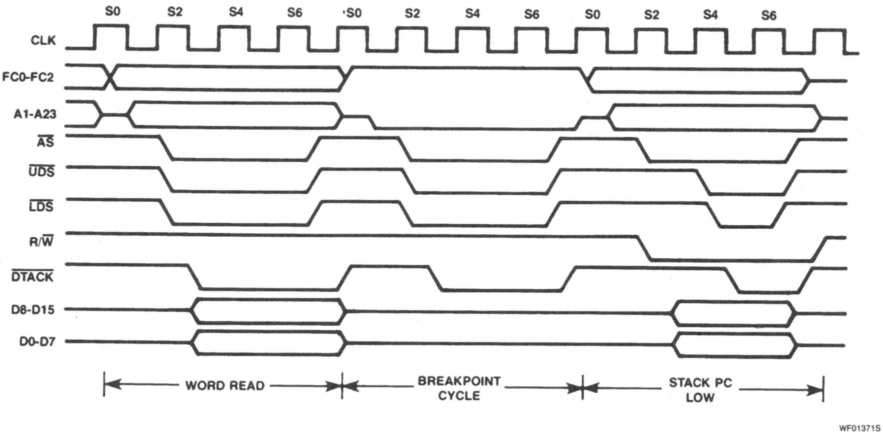


Figure 39. Breakpoint Cycle Timing Diagram

Priority level seven is a special case. Level seven interrupts cannot be inhibited by the interrupt priority mask, thus providing a "non-maskable interrupt" capability. An interrupt is generated each time the interrupt request level changes from some lower level to level seven. Note that a level seven interrupt may still be caused by the level comparison if the request level is a seven and the processor priority is set to a lower level by an instruction.

Uninitialized Interrupt

An interrupting device asserts $\overline{\text{VPA}}$, $\overline{\text{BERR}}$, or provides an interrupt vector number and asserts $\overline{\text{DTACK}}$ during an interrupt acknowledge cycle by the SCN68010. If the vector register has not been initialized, the responding S68000 family peripheral will provide vector number 15, the uninitialized interrupt vector. This provides a uniform way to recover from a programming error.

Spurious Interrupt

If during the interrupt acknowledge cycle no device responds by asserting $\overline{\text{DTACK}}$ or $\overline{\text{VPA}}$, $\overline{\text{BERR}}$ should be asserted to terminate the vector acquisition. The processor separates the processing of this error from bus error by forming a short format exception stack and fetching the spurious interrupt vector instead of the bus error vector. The processor then proceeds with the usual exception processing.

Instruction Traps

Traps are exceptions caused by instructions. They arise either from processor recognition of abnormal conditions during instruction execution, or from use of instructions whose normal behavior is trapping.

Some instructions are used specifically to generate traps. The TRAP instruction always forces an exception and is useful for implementing supervisor calls for user programs. The TRAPV and CHK instructions force an exception if the user program detects a runtime error, which may be an arithmetic overflow or a subscript out of bounds.

The signed divide (DIVS) and unsigned divide (DIVU) instructions will force an exception if a division operation is attempted with a divisor of zero.

Illegal and Unimplemented Instructions

Illegal instruction is the term used to refer to any of the word bit patterns which are not the bit patterns of the first word of a legal SCN68010 instruction. During instruction execution, if such an instruction is fetched, an illegal instruction exception occurs. Signetics reserves the right to define instructions whose opcodes may be any of the illegal instructions. Three bit patterns will always force an illegal instruction trap on all S68000 family compatible microprocessors. They are: $4AFA, $4AFB, and $4AFC. Two of the patterns, $4AFA and $4AFB, are reserved for Signetics system products. The third pattern, $4AFC, is reserved for customer use.

In addition to the previously defined illegal instruction opcodes, the SCN68010 defines eight breakpoint illegal instructions with the bit patterns $4848 – $484F. These instructions cause the processor to enter illegal instruction exception processing as usual, but a breakpoint bus cycle is executed before the stacking operations are performed as shown in Figure 39. The processor does not accept or send any data during this cycle. Whether the breakpoint cycle is terminated with a $\overline{\text{DTACK}}$, $\overline{\text{BERR}}$, or $\overline{\text{VPA}}$ signal, the processor will continue with the illegal instruction processing. The purpose of this cycle is to provide a software breakpoint that will signal external hardware when it is executed.

16-Bit Virtual Memory Microprocessor

SCN68010

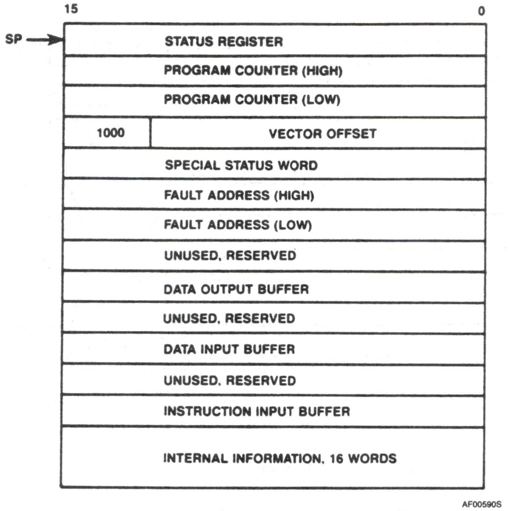


Figure 40. Exception Stack Order (Bus and Address Error)

Word patterns with bits 12 – 15 equaling 1010 or 1111 are distinguished as unimplemented instructions and separate exception vectors are given to these patterns to permit efficient emulation. This facility allows the operating system to detect program errors, or to emulate unimplemented instructions, such as the 68881 floating-point coprocessor instructions, in software.

Privilege Violations

In order to provide system security, various instructions are privileged. An attempt to execute one of the privileged instructions while in the user state will cause an exception. The privileged instructions are:

AND immediate to SR
EOR immediate to SR
MOVE to SR
MOVE from SR
MOVEC
MOVES
MOVE USP
OR immediate to SR
RESET
RTE
STOP

Tracing

To aid in program development, the SCN68010 includes a facility to allow instruction-by-instruction tracing. In the trace state, after each instruction is executed an exception is forced, allowing a debugging program to monitor the execution of the program under test.

The trace facility uses the T bit in the supervisor portion of the status register. If the T bit is negated (off), tracing is disabled, and instruction execution proceeds from instruction to instruction as normal. If the T bit is asserted (on) at the beginning of the execution of an instruction, a trace exception will be generated as the execution of that instruction is completed. If the instruction is not executed, either because an interrupt is taken, or the instruction is illegal or privileged, the trace exception does not occur. The trace exception also does not occur if the instruction is aborted by a reset, bus error, or address error exception. If the instruction is indeed executed and an interrupt is pending on completion, the trace exception is processed before the interrupt exception. If, during the execution of the instruction an exception is forced by that instruction, the forced exception is processed before the trace exception.

As an extreme illustration of the above rules, consider the arrival of an interrupt during the execution of a TRAP instruction while tracing is enabled. First the trap exception is processed, then the trace exception, and finally the interrupt exception. Instruction execution resumes in the interrupt handler routine.

Bus Error

Bus error exceptions occur when external logic terminates a bus cycle with a bus error signal. Whether the processor was doing instruction or exception processing, that processing is terminated, and the processor immediately begins exception processing. However, if a bus error occurs during exception processing for a bus error, address error, or reset, the processor detects a double bus fault and halts. When exception processing is completed, instruction execution continues at the address contained in exception vector table entry two, at offset $008.

Exception processing for a bus error follows a slightly different sequence than the sequence for group 1 and 2 exceptions. In addition to the four steps executed during exception processing for all other exceptions, 22 words of additional information are placed on the stack. This additional information describes the internal state of the processor at the time of the bus error and is reloaded by the RTE instruction to continue the instruction that caused the error. Figure 40 shows the order of the stacked information.

The value of the saved program counter does not necessarily point to the instruction that was executing when the bus error occurred, but may be advanced by up to five words. This is due to the prefetch mechanism on the SCN68010 that always fetches a new instruction word as each previously fetched instruction word is used (see **Instruction Prefetch**). However, enough information is placed on the stack for the bus error exception handler routine to determine why the bus fault occurred. This additional information includes the address that was being accessed, the function codes for the access, whether it was a read or a write, and what internal register was included in the transfer. The fault address can be used by an operating system to determine what virtual memory location is needed so that the requested data can be brought into physical memory. The RTE instruction is then used to reload the processor's internal state at the time of the fault, the faulted bus cycle will then be rerun and the suspended instruction completed. If the faulted bus cycle was a read-modify-write, the entire cycle will be rerun whether the fault occurred during the read or the write operation.

An alternate method of handling a bus error is to complete the faulted access in software. In order to use this method, use of the special status word, the instruction input buffer, the data input buffer, and the data output buffer image is required. The format of the special status word is shown in Figure 41. If the bus cycle was a write, the data output buffer image should be written to the fault address location using the function code contained in the special status word. If the cycle was a read, the data at the fault address location should be written to the images of the data input buffer, instruction input buffer, or both according to the DF and IF bits.[1] In addition, for read-modify-write cycles, the status register image must be properly set to reflect the read data if the fault occurred during the read portion of the cycle and the write operation (i.e., setting the most significant bit of the memory location) must also be performed. This is because the entire read-modify-write cycle is assumed to have been completed by software. Once the cycle has been completed by software, the RR bit in the special status word is set to indicate to the processor that it should not rerun the cycle when the RTE instruction is executed. If the rerun flag is set when an RTE instruction executes, the SCN68010 still reads all of the information from the stack.

---

[1] If the faulted access was a byte operation, the data should be moved from or to the least-significant byte of the data output or input buffer images unless the HB bit is set. This condition will only occur if a MOVEP instruction caused the fault during the transfer of bits 8 – 15 of a word or long word or bits 24 – 31 of a long word.

15	14	13	12	11	10	9	8	7 – 3	2	1	0
RR	*	IF	DF	RM	HB	BY	RW	*	FC2-FC0		

RR – RE-RUN FLAG; 0 = PROCESSOR RE-RUN (DEFAULT), 1 = SOFTWARE RE-RUN.
IF – INSTRUCTION FETCH TO THE INSTRUCTION INPUT BUFFER.
DF – DATA FETCH TO THE DATA INPUT BUFFER.
RM – READ-MODIFY-WRITE CYCLE.
HB – HIGH BYTE TRANSFER FROM THE DATA OUTPUT BUFFER OR TO THE DATA INPUT BUFFER.
BY – BYTE TRANSFER FLAG; HB SELECTS THE HIGH OR LOW BYTE OF THE TRANSFER REGISTER. IF BY IS CLEAR, THE TRANSFER IS WORD.
RW – READ/WRITE FLAG; 0 = WRITE, 1 = READ.
FC – THE FUNCTION CODE USED DURING THE FAULTED ACCESS.
* – THESE BITS ARE RESERVED FOR FUTURE USE BY SIGNETICS AND WILL BE ZERO WHEN WRITTEN BY THE SCN68010.

AF00600S

Figure 41. Special Status Word Format

Address Error

Address error exceptions occur when the processor attempts to access a word or long word operand or an instruction at an odd address. The effect is much like an internally generated bus error, so that the bus cycle is aborted, and the processor begins exception processing. After exception processing commences, the sequence is the same as that for bus error including the information that is stacked, except that the vector offset refers to the address error exception vector. If an address error occurs during exception processing for a bus error, address error, or reset, the processor detects a double bus fault and halts.

As shown in Figure 42, an address error will execute a short bus cycle followed by exception processing. This short bus cycle is similar to a normal read or write cycle, except that the data strobes are not asserted and no external signals are used to terminate the cycle. During an address error bus cycle, AS is asserted to indicate that the SCN68010 will drive the address bus (thus allowing for proper operation in a multiple bus master system). Note that data strobes are not asserted allowing for address error detection and memory protection.

Since the address error exception stacks the same information that is stacked by a bus error exception, it is possible to use the RTE instruction to continue execution of the suspended instruction. However, if the software rerun flag is not set, the fault address will be used when the cycle is rerun and another address error exception will occur. Therefore, the user must be certain that the proper corrections have been made to the stack image and user registers before attempting to continue the instruction. With proper software handling, the address error exception handler could emulate word or long word accesses to odd addresses if desired.

Return From Exception

In addition to returning from any exception handler routine, the RTE instruction is used to resume the execution of a suspended instruction by restoring all of the temporary register and control information stored during a bus error and returning to the normal processing state. For the RTE instruction to execute properly, the stack must contain valid and accessible data. The RTE instruction checks for data validity in two ways; first, by checking the format/offset word for a valid stack format code, and second, if the format code indicates the long stack format, the long stack data is checked for validity as it is loaded into the processor. In addition, the data is checked for accessibility when the processor starts reading the long data. Because of these checks, the RTE instruction executes as follows:

1. Determine the stack format. This step is the same for any stack format and consists of reading the status register, program counter, and format/offset word. If the format code indicates a short stack format, execution continues at the new program counter address. If the format code is not one of the SCN68010 defined stack format codes, exception processing starts for a format error.
2. Determine data validity. For a long stack format, the SCN68010 will begin to read the remaining stack data, checking for validity of the data. The only word checked for validity is the first of the 16 internal information words (SP + 26)

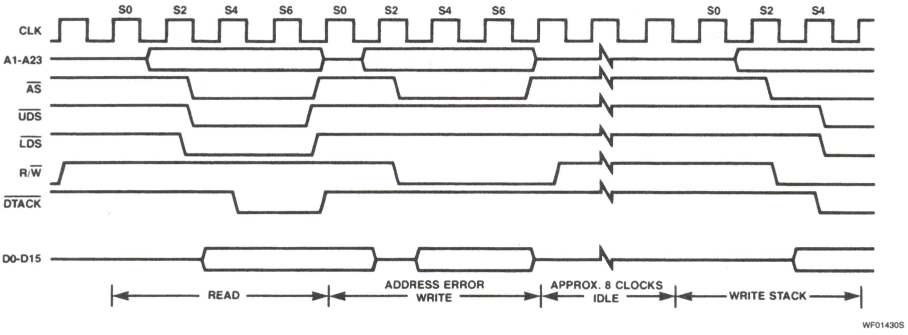


Figure 42. Address Error Timing Diagram

shown in Figure 40. This word contains a processor version number in bits 10 through 13, that must match the version number of the SCN68010 that is attempting to read the data. This validity check is used to insure that in dual processor systems, the data will be properly interpreted by the RTE instruction if the two processors are of different versions. If the version number is incorrect for this processor, the RTE instruction will be aborted and exception processing will begin for a format error exception. Since the stack pointer is not updated until the RTE instruction has successfully read all of the stack data, a format error occurring at this point will not stack new data over the previous bus error stack information.

3. Determine data accessibility. If the long stack data is valid, the SCN68010 performs a read from the last word (SP + 56) of the long stack to determine data accessibility. If this read is terminated normally, the processor assumes that the remaining words on the stack frame are also accessible. If a bus error is signaled before or during this read, a bus error exception is taken as usual. After this read, the processor must be able to load the remaining data without receiving a bus error; therefore, if a bus error occurs on any of the remaining stack reads, the SCN68010 treats this as a double bus fault and enters the halted state.

INTERFACE WITH SYNCHRONOUS PERIPHERALS

To interface the synchronous peripherals with the asynchronous SCN68010, the processor modifies its bus cycle to meet the synchronous cycle requirements whenever a synchronous device address is detected. This is possible since both processors use memory mapped I/O. Figure 43 is a flowchart of the interface operation between the processor and synchronous devices.

Data Transfer Operation

Three signals on the processor provide the synchronous interface. They are: enable (E), valid memory address ($\overline{\text{VMA}}$), and valid peripheral address ($\overline{\text{VPA}}$). Enable corresponds to the E or phase 2 signal in existing synchronous systems. The bus frequency is one tenth of the incoming SCN68010 clock frequency. The timing of E allows 1MHz peripherals to be used with an 8MHz SCN68010. Enable has a 60/40 duty cycle; that is, it is low for six input clocks and high for four input clocks. This duty cycle allows the processor to do successive $\overline{\text{VPA}}$ accesses on successive E pulses.

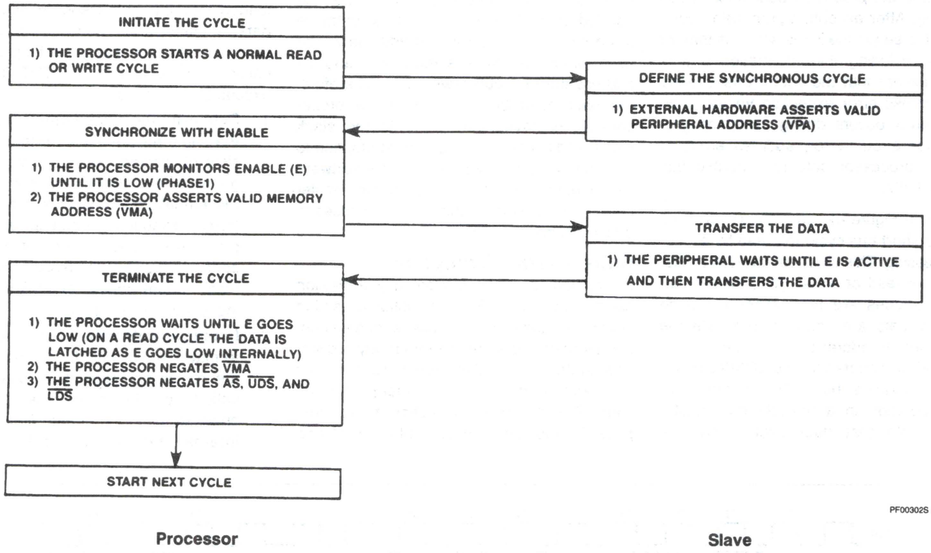

INITIATE THE CYCLE
1) THE PROCESSOR STARTS A NORMAL READ OR WRITE CYCLE
DEFINE THE SYNCHRONOUS CYCLE
1) EXTERNAL HARDWARE ASSERTS VALID PERIPHERAL ADDRESS (VPA)
SYNCHRONIZE WITH ENABLE
1) THE PROCESSOR MONITORS ENABLE (E) UNTIL IT IS LOW (PHASE1)
2) THE PROCESSOR ASSERTS VALID MEMORY ADDRESS (VMA)
TRANSFER THE DATA
1) THE PERIPHERAL WAITS UNTIL E IS ACTIVE AND THEN TRANSFERS THE DATA
TERMINATE THE CYCLE
1) THE PROCESSOR WAITS UNTIL E GOES LOW (ON A READ CYCLE THE DATA IS LATCHED AS E GOES LOW INTERNALLY)
2) THE PROCESSOR NEGATES VMA
3) THE PROCESSOR NEGATES AS, UDS, AND LDS
START NEXT CYCLE
PF00302S
Processor
Slave

Figure 43. Synchronous Interfacing Flowchart

Synchronous cycle timing is given in Figure 44. At state zero (S0) in the cycle, the address bus is in the high-impedance state. A function code is asserted on the function code output lines. One-half clock later, in state 1, the address bus is released from the high-impedance state.

During state 2, the address strobe ($\overline{\mathrm{AS}}$) is asserted to indicate that there is a valid address on the address bus. If the bus cycle is a read cycle, the upper and/or lower data strobes are also asserted in state 2. If the bus cycle is a write cycle, the read/write (R/$\overline{\mathrm{W}}$) signal is switched to low (write) during state 2. One-half clock later, in state 3, the write data is placed on the data bus, and in state 4 the data strobes are issued to indicate valid data on the data bus. The processor now inserts wait states until it recognizes the assertion of $\overline{\mathrm{VPA}}$.

The $\overline{\mathrm{VPA}}$ input signals the processor that the address on the bus is the address of a synchronous device (or an area reserved for synchronous devices) and that the bus should conform to the phase 2 transfer characteristics of the synchronous bus. Valid peripheral address is derived by decoding the address bus, conditioned by the address strobe. Chip select for the synchronous peripherals should be derived by decoding the address bus conditioned by $\overline{\mathrm{VMA}}$.

After recognition of $\overline{\mathrm{VPA}}$, the processor assures that the enable (E) is low, by waiting if necessary, and subsequently asserts $\overline{\mathrm{VMA}}$ two clock cycles before E goes high. $\overline{\mathrm{VMA}}$ is then used as part of the chip select equation of the peripheral. This ensures that the synchronous peripherals are selected and deselected at the correct time. The peripheral now runs its cycle during the high portion of the E signal. Figures 44 and 45 depict the best and worst case synchronous cycle timing; this cycle length is dependent strictly upon when $\overline{\mathrm{VPA}}$ is asserted in relationship to the E clock.

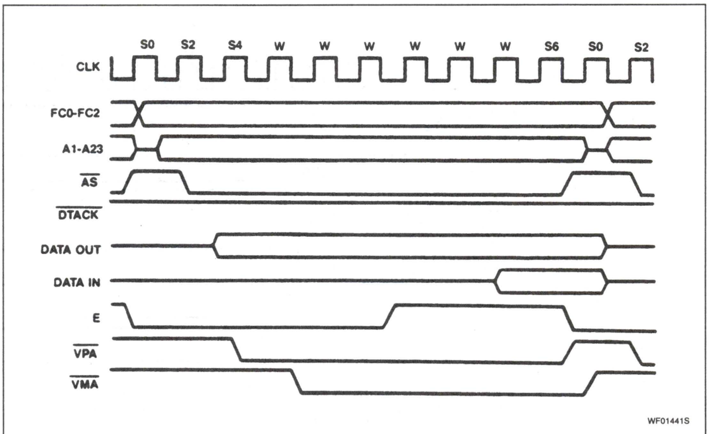


Figure 44. SCN68010 to Synchronous Peripheral Timing Diagram — Best Case

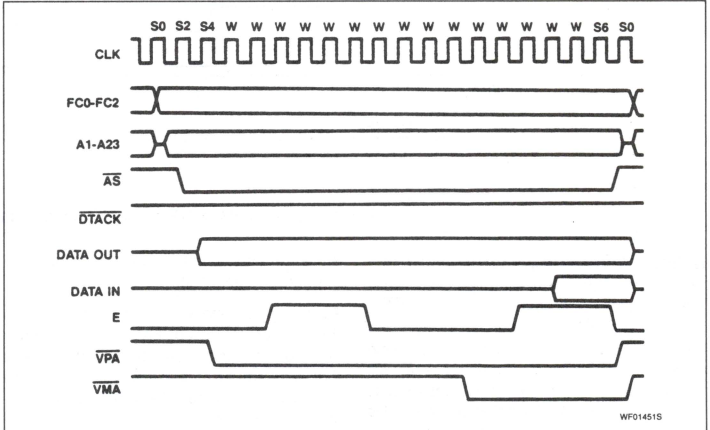


Figure 45. SCN68010 to Synchronous Peripheral Timing Diagram — Worst Case

If we assume that external circuitry asserts $\overline{\mathrm{VPA}}$ as soon as possible after the assertion of $\overline{\mathrm{AS}}$, then $\overline{\mathrm{VPA}}$ will be recognized as being asserted on the falling edge of S4. In this case, no "extra" wait cycles will be inserted prior to the recognition of $\overline{\mathrm{VPA}}$ asserted and only the wait cycles inserted to synchronize with the E clock will determine the total length of the cycle. In any case, the synchronization delay will be some integral number of clock cycles within the following two extremes:

1. Best Case — $\overline{\mathrm{VPA}}$ is recognized as being asserted on the falling edge three clock cycles before E rises (or three clock cycles after E falls).
2. Worst Case — $\overline{\mathrm{VPA}}$ is recognized as being asserted on the falling edge two clock cycles before E rises (or four clock cycles after E falls).

During a read cycle, the processor latches the peripheral data in state 6. For all cycles, the processor negates the address and data strobes one-half clock cycle later in state 7 and the enable signal goes low at this time. Another half clock later, the address bus is put in the high-impedance state. During a write cycle, the data bus is put in the high-impedance state and the read/write signal is switched high. The peripheral logic must remove $\overline{\mathrm{VPA}}$ within one clock after the address strobe is negated.

$\overline{\mathrm{DTACK}}$ should not be asserted while $\overline{\mathrm{VPA}}$ is asserted. Notice that the SCN68010 $\overline{\mathrm{VMA}}$ is active low, contrasted with the active high synchronous VMA. This allows the processor to put its buses in the high-impedance state on DMA requests without inadvertently selecting the peripherals.

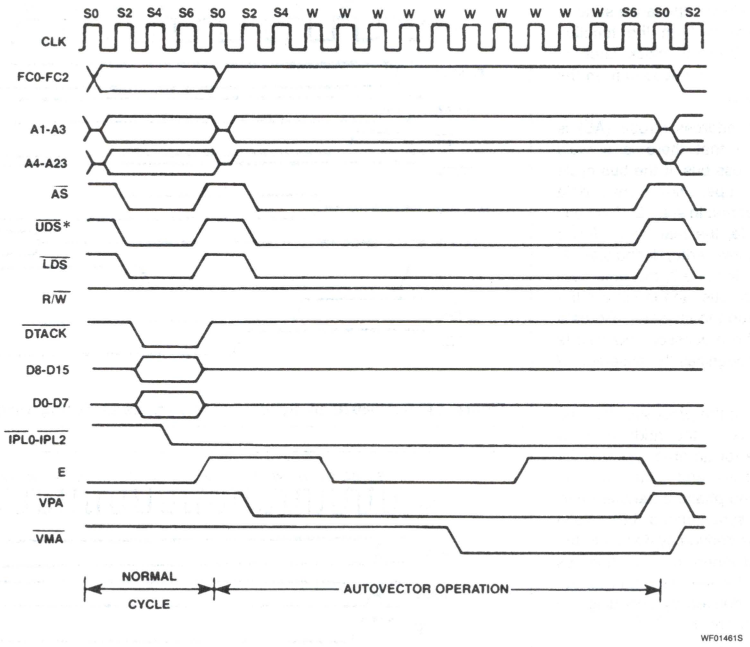

S0 S2 S4 S6 S0 S2 S4 W W W W W W W W W W S6 S0 S2
CLK
FC0-FC2
A1-A3
A4-A23
AS
UDS*
LDS
R/W
DTACK
D8-D15
D0-D7
IPL0-IPL2
E
VPA
VMA
NORMAL CYCLE
AUTOVECTOR OPERATION
WF01461S

Figure 46. Autovector Operation Timing Diagram

Interrupt Interface Operation

During an interrupt acknowledge cycle while the processor is fetching the vector, if $\overline{VPA}$ is asserted the SCN68010 will assert $\overline{VMA}$ and complete a normal synchronous read cycle as shown in Figure 46. The processor will then use an internally generated vector that is a function of the interrupt being serviced. This process is known as autovectoring. The seven autovectors are vector numbers 25 through 31 (decimal).

Autovectoring operates in the same fashion (but is not restricted to) the synchronous interrupt sequence. The basic difference is that there are six normal interrupt vectors and one NMI type vector. As with both the synchronous and the SCN68010's normal vectored interrupt, the interrupt service routine can be located anywhere in the address space. This is due to the fact that while the vector numbers are fixed, the contents of the vector table entries are assigned by the user.

Since $\overline{VMA}$ is asserted during autovectoring, the synchronous peripheral address decoding should prevent unintended accesses.

INSTRUCTION SET AND EXECUTION TIMES

Instruction Set

The following paragraphs provide information about the addressing categories and instruction set of the SCN68010.

Addressing Categories

Effective address modes may be categorized by the ways in which they may be used. The following classifications will be used in the instruction definitions:

Data	If an effective address mode may be used to refer to data operands, it is considered a data addressing effective address mode.
Memory	If an effective address mode may be used to refer to memory operands, it is considered a memory addressing effective address mode.
Alterable	If an effective address mode may be used to refer to alterable (writeable) operands, it is considered an alterable addressing effective address mode.
Control	If an effective address mode may be used to refer to memory operands without an associated size, it is considered a control addressing effective address mode.

Table 22. Effective Addressing Mode Categories

EFFECTIVE ADDRESS MODES	MODE	REGISTER	DATA	ADDRESSING CATEGORIES		
				Memory	Control	Alterable
Dn	000	Register number	X	–	–	X
An	001	Register number	–	–	–	X
(An)	010	Register number	X	X	X	X
(An)+	011	Register number	X	X	–	X
–(An)	100	Register number	X	X	–	X
d(An)	101	Register number	X	X	X	X
d(An, ix)	110	Register number	X	X	X	X
xxx.W	111	000	X	X	X	X
xxx.L	111	001	X	X	X	X
d(PC)	111	010	X	X	X	–
d(PC, ix)	111	011	X	X	X	–
#xxx	111	100	X	X	–	–

These categories may be combined, so that additional, more restrictive, classifications may be defined. For example, the instruction descriptions use such classifications as alterable memory or data alterable. The former refers to those addressing modes which are both alterable and memory addresses, and the latter refers to addressing modes which are both data and alterable. Table 22 shows the various categories to which each of the effective address modes belong. Table 23 is the instruction set summary.

Table 23. Instruction Set

MNEMONIC	DESCRIPTION	OPERATION	CONDITION CODES				
			X	N	Z	V	C
ABCD	Add decimal with extend	(Destination)$_{10}$ + (Source)$_{10}$ + X → Destination	*	U	*	U	*
ADD	Add binary	(Destination) + (Source) → Destination	*	*	*	*	*
ADDA	Add address	(Destination) + (Source) → Destination	–	–	–	–	–
ADDI	Add immediate	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDQ	Add quick	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDX	Add extended	(Destination) + (Source) + X= Destination	*	*	*	*	*
AND	AND logical	(Destination) ∧ Source → Destination	–	*	*	0	0
ANDI	AND immediate	(Destination) ∧ Immediate Data → Destination	–	*	*	0	0
ANDI to CCR	AND immediate to condition codes	(Source) ∧ CCR → CCR	*	*	*	*	*
ANDI to SR	AND immediate to status register	(Source) ∧ SR → SR	*	*	*	*	*
ASL, ASR	Arithmetic shift	(Destination) Shifted by < count > → Destination	*	*	*	*	*
B$_{CC}$	Branch conditionally	If $_{CC}$ the PC + d → PC	–	–	–	–	–
BCHG	Test a bit and change	~(< bit number >) OF Destination → Z ~(< bit number >) OF Destination → < bit number > OF Destination	–	–	*	–	–
BCLR	Test a bit and clear	~(< bit number >) OF Destination → Z 0 → < bit number > → OF Destination	–	–	*	–	–
BRA	Branch always	(PC) + displacement → PC	–	–	–	–	–
BSET	Test a bit and set	~(< bit number >) OF Destination → Z 1 → < bit number > OF Destination	–	–	*	–	–
BSR	Branch to subroutine	(PC) → –(SP); (PC) + d → PC	–	–	–	–	–
BTST	Test a bit	~(< bit number >) OF Destination → Z	–	–	*	–	–
CHK	Check register against bounds	If Dn < 0 or Dn > (< ea >) then TRAP	–	*	U	U	U
CLR	Clear and operand	0 → Destination	–	0	1	0	0
CMP	Compare	(Destination) – (Source)	–	*	*	*	*
CMPA	Compare address	(Destination) – (Source)	–	*	*	*	*
CMPI	Compare immediate	(Destination) – Immediate Data	–	*	*	*	*
CMPM	Compare memory	(Destination) – (Source)	–	*	*	*	*
DB$_{CC}$	Test condition, decrement and branch	if ~$_{CC}$ then Dn – 1 → Dn; if Dn ≠ –1 then PC + d → PC	–	–	–	–	–
DIVS	Signed divide	(Destination)/(Source) → Destination	–	*	*	*	0
DIVU	Unsigned divide	(Destination)/(Source) → Destination	–	*	*	*	0
EOR	Exclusive OR logical	(Destination) ⊕ (Source) → Destination	–	*	*	0	0
EORI	Exclusive OR immediate	(Destination) ⊕ Immediate Data → Destination	–	*	*	0	0
EORI to CCR	Exclusive OR immediate to condition codes	(Source) ⊕ CCR → CCR	*	*	*	*	*
EORI to SR	Exclusive OR immediate to status register	(Source) ⊕ SR → SR	*	*	*	*	*
EXG	Exchange register	(Rx) ↔ (Ry)	–	–	–	–	–
EXT	Sign extend	(Destination) Sign-Extended → Destination	–	*	*	0	0
JMP	Jump	(Destination) → PC	–	–	–	–	–
JSR	Jump to subroutine	(PC) → – (SP); Destination → PC	–	–	–	–	–
LEA	Load effective address	Destination → An	–	–	–	–	–
LINK	Link and allocate	(An) → – (SP); (SP) → An; (SP) + → SP	–	–	–	–	–
LSL, LSR	Logical shift	(Destination) Shifted by < count > → Destination	*	*	*	0	*
MOVE	Move data from source to destination	(Source) → Destination	–	*	*	0	0
MOVE to CCR	Move to condition code	(Source) → CCR	*	*	*	*	*
MOVE from CCR	Move from condition codes	(CCR) → Destination	–	–	–	–	–
MOVE to SR	Move to the status register	(Source) → SR	*	*	*	*	*
MOVE from SR	Move from the status register	(SR) → Destination	–	–	–	–	–
MOVE USP	Move user stack pointer	(USP) → An; (An)=Ar USP	–	–	–	–	–
MOVEA	Move address	(Source) → Destination	–	–	–	–	–
MOVEC	Move control register	(Cr) → Rn; (Rn) → Cr	–	–	–	–	–

Table 23. Instruction Set (Continued)

MNEMONIC	DESCRIPTION	OPERATION	CONDITION CODES				
			X	N	Z	V	C
MOVEM	Move multiple registers	(Registers) → Destination (Source) → Registers	–	–	–	–	–
MOVEP	Move peripheral data	(Source) → Destination	–	–	–	–	–
MOVEQ	Move quick	Immediate Data → Destination	–	*	*	0	0
MOVES	Move alternate address space	(Dn) → Destination; (Source) → Dn	–	–	–	–	–
MULS	Signed multiply	(Destination)X (Source) → Destination	–	*	*	0	0
MULU	Unsigned multiply	(Destination)X (Source) → Destination	–	*	*	0	0
NBCD	Negate decimal with extend	0 – (Destination)$_{10}$ – X → Destination	*	U	*	U	*
NEG	Negate	0 – (Destination) → Destination	*	*	*	*	*
NEGX	Negate with extend	0 – (Destination) – X → Destination	*	*	*	*	*
NOP	No operation	–	–	–	–	–	–
NOT	Logical complement	~ (Destination) → Destination	–	*	*	0	0
OR	Inclusive OR logical	(Destination) v (Source) → Destination	–	*	*	0	0
ORI	Inclusive OR immediate	(Destination) v Immediate Data → Destination	–	*	*	0	0
ORI to CCR	Inclusive OR immediate to condition codes	(Source) V CCR → CCR	*	*	*	*	*
ORI to SR	Inclusive OR immediate to status register	(Source) V SR → SR	*	*	*	*	*
PEA	Push effective address	Destination → – (SP)	–	–	–	–	–
RESET	Reset external device	–	–	–	–	–	–
ROL, ROR	Rotate (without extend)	(Destination) Rotated by < count > → Destination	–	*	*	0	*
ROXL, ROXR	Rotate with extend	(Destination) Rotated by < count > → Destination	*	*	*	0	*
RTD	Return and deallocate stack	(SP)+ → PC; (SP) + d → SP	–	–	–	–	–
RTE	Return from exception	(SP) + → SR; (SP) + → PC	*	*	*	*	*
RTR	Return and restore condition codes	(SP) + → CC; (SP) + → PC	*	*	*	*	*
RTS	Return from subroutine	(SP) + → PC	–	–	–	–	–
SBCD	Subtract decimal with extend	(Destination)$_{10}$ – (Source) $_{10}$ – X → Destination	*	U	*	U	*
S$_{CC}$	Set according to condition	If $_{CC}$ then 1's → Destination else 0's → Destination	–	–	–	–	–
STOP	Load status register and stop	Immediate Data → SR; STOP	*	*	*	*	*
SUB	Subtract binary	(Destination) – (Source) → Destination	*	*	*	*	*
SUBA	Subtract address	(Destination) – (Source) → Destination	–	–	–	–	–
SUBI	Subtract immediate	(Destination) – Immediate Data → Destination	*	*	*	*	*
SUBQ	Subtract quick	(Destination) – Immediate Data → Destination	*	*	*	*	*
SUBX	Subtract with extend	(Destination) – (Source) – X → Destination	*	*	*	*	*
SWAP	Swap register halves	Register [31:16] ↔ Register [15:0]	–	*	*	0	0
TAS	Test and set an operand	(Destination) Tested → CC; 1 → [7] OF Destination	–	*	*	0	0
TRAP	Trap	(PC) → – (SSP); (SR) → – (SSP); (Vector) → PC	–	–	–	–	–
TRAPV	Trap on overflow	If V set then TRAP	–	–	–	–	–
TST	Test and operand	(Destination) Tested → CC	–	*	*	0	0
UNLK	Unlink	(An) → SP; (SP) + → An	–	–	–	–	–

NOTES:

[ ] = bit number
⊕ logical exclusive OR
∧ logical AND
V logical OR
~ logical complement

* affected
– unaffected
0 cleared
1 set
U undefined

Instruction Prefetch

The SCN68010 uses a two-word tightly-coupled instruction prefetch mechanism to enhance performance. This mechanism is described in terms of the microcode operations involved. If the execution of an instruction is defined to begin when the microroutine for that instruction is entered, some features of the prefetch mechanism can be described.

1. When execution of an instruction begins, the operation word and the word following have already been fetched. The operation word is in the instruction decoder.
2. In the case of multi-word instructions, as each additional word of the instruction is used internally, a fetch is made to the instruction stream to replace it.
3. The last fetch for an instruction from the instruction stream is made when the operation word is discarded and decoding is started on the next instruction.
4. If the instruction is a single-word instruction causing a branch, the second word is not used. But because this word is fetched by the preceding instruction, it is impossible to avoid this superfluous fetch.
5. In the case of an interrupt or trace exception, both prefetched words are not used.
6. The program counter usually points to the last word fetched from the instruction stream.

Loop Mode Operation

The SCN68010 has several features that provide efficient execution of program loops. One of these features is the DBcc looping primitive instruction. The DBcc instruction operates on three operands, a loop counter, a branch condition, and a branch displacement. When the DBcc is executed in loop mode, the contents of the low order word of the register specified as the loop counter is decremented by one and compared to minus one. If equal to minus one, the result of the decrement is placed back into the count register and the next sequential instruction is executed, otherwise the condition code register is checked against the specified branch condition. If the condition is true, the result of the decrement is discarded and the next sequential instruction is executed. Finally, if the count register is not equal to minus one and the branch condition is false, the branch displacement is added to the program counter and instruction execution continues at that new address. Note that this is slightly different than non-looped execution; however, the results are the same.

```
      LEA      SOURCE, A0       Load a pointer to source data
      LEA      DEST, A1         Load a pointer to destination
      MOVE. W  #LENGTH, D0      Load the counter register
LOOP  MOVE. W  (A0)+, (A1)+     Loop to move the block of data
      DBEQ     D0, LOOP         Stop if data word is zero
```

Figure 47. DBcc Loop Program Example

An example of using the DBcc instruction in a simple loop for moving a block of data is shown in Figure 47. In this program, the block of data 'LENGTH' words long at address 'SOURCE' is to be moved to address 'DEST' provided that none of the words moved are equal to zero. When the effect of instruction prefetch on this loop is examined it can be seen that the bus activity during the loop execution would be:

1. Fetch the MOVE.W instruction,
2. Fetch the DBEQ instruction,
3. Read the operand where A0 points,
4. Write the operand where A1 points,
5. Fetch the DBEQ branch displacement, and
6. If loop conditions are met, return to step 1.

During this loop, five bus cycles are executed; however, only two bus cycles perform the data movement. Since the SCN68010 has a two word prefetch queue in addition to a one word instruction decode register, it is evident that the three instruction fetches in this loop could be eliminated by placing the MOVE.W word in the instruction decode register and holding the DBEQ instruction and its branch displacement in the prefetch queue. The SCN68010 has the ability to do this by entering the loop mode of operation. During loop mode operation, all opcode fetches are suppressed and only operand reads and writes are performed until an exit loop condition is met.

Loop mode operation is transparent to the programmer, with only two conditions required for the SCN68010 to enter the loop mode. First, a DBcc instruction must be executed with both branch conditions met and a branch displacement of minus four; which indicates that the branch is to a one word instruction preceding the DBcc instruction. Second, when the processor fetches the instruction at the branch address, it is checked to determine whether it is one of the allowed looping instructions. If it is, the loop mode is entered. Thus, the single word looped instruction and the first word of the DBcc instruction will each be fetched twice when the loop is entered; but no instruction fetches will occur again until the DBcc loop conditions fail.

In addition to the normal termination conditions for a loop, there are several conditions that will cause the SCN68010 to exit loop mode operation. These conditions are interrupts, trace exceptions, reset errors, and bus errors. Interrupts are honored after each execution of the DBcc instruction, but not after the execution of the looped instruction. If an interrupt exception occurs, loop mode operation is terminated and can be restarted on return from the interrupt handler. If the T bit is set, trace exceptions will occur at the end of both the loop instruction and the DBcc instruction and thus loop mode operation is not available. Reset will abort all processing, including the loop mode. Bus errors during the loop mode will be treated the same as in normal processing; however, when the RTE instruction is used to continue the execution of the looped instruction, the three word loop will not be re-fetched.

The loopable instructions available on the SCN68010 are listed in Table 24. These instructions may use the three address register indirect addressing modes to form one word looping instructions; (An), (An) + , and –(An).

Table 24. MC68010 Loopable Instructions

OPCODES	APPLICABLE ADDRESSING MODES
MOVE [BWL]	(Ay) to (Ax) (Ay) to (Ax)+ (Ay) to -(Ax) (Ay)+ to (Ax) (Ay)+ to (Ax)+ (Ay)+ to -(Ax) -(Ay) to (Ax) -(Ay) to (Ax)+ -(Ay) to -(Ax) Ry to (Ax) Ry to (Ax)+
ADD [BWL] AND [BWL] CMP [BWL] OR [BWL] SUB [BWL]	(Ay) to Dx (Ay)+ to Dx -(Ay) to Dx
ADDA [WL] CMPA [WL] SUBA [WL]	(Ay) to Ax -(Ay) to Ax (Ay)+ to Ax
ADD [BWL] AND [BWL] EOR [BWL] OR [BWL] SUB [BWL]	Dx to (Ay) Dx to (Ay)+ Dx to -(Ay)
ABCD [B] ADDX [BWL] SBCD [B] SUBX [BWL]	-(Ay) to -(Ax)
CMP [BWL]	(Ay)+ to (Ax)+
CLR [BWL] NEG [BWL] NEGX [BWL] NOT [BWL] TST [BWL] NBCD [B]	(Ay) (Ay)+ -(Ay)
ASL [W] ASR [W] LSL [W] LSR [W] ROL [W] ROR [W] ROXL [W] ROXR [W]	(Ay) by #1 (Ay)+ by #1 -(Ay) by #1

NOTE:
[B, W, or L] indicate an operand size of byte, word, or long word.

Instruction Execution Times

The following paragraphs contain listings of the instruction execution times in terms of external clock (CLK) periods. In this timing data, it is assumed that both memory read and write cycle times are four clock periods. Any wait states caused by a longer memory cycle must be added to the total instruction time. The number of bus read and write cycles for each instruction is also included with the timing data. This data is enclosed in parenthesis following the execution periods and is shown as (r/w) where r is the number of read cycles and w is the number of write cycles.

NOTE

The number of clock periods includes instruction fetches and all applicable operand fetches and stores.

Operand Effective Address Calculation Times

Table 25 lists the number of clock periods required to compute an instruction's effective address. It includes fetching of any extension words, the address computation, and fetching of the memory operand if necessary. Several instructions do not need the operand at an effective address to be fetched and thus require fewer clock periods to calculate a given effective address than the instructions that do fetch the effective address operand. The number of bus read and write cycles is shown in parentheses as (r/w). Note there are no write cycles involved in processing the effective address.

Table 25. Effective Address Calculation Times

ADDRESSING MODE		BYTE, WORD		LONG	
		Fetch	No Fetch	Fetch	No Fetch
Register					
Dn	Data register direct	0(0/0)	–	0(0/0)	–
An	Address register direct	0(0/0)	–	0(0/0)	–
Memory					
(An)	Address register indirect	4(1/0)	2(0/0)	8(2/0)	2(0/0)
(An)+	Address register indirect with postincrement	4(1/0)	4(0/0)	8(2/0)	4(0/0)
-(An)	Address register indirect with predecrement	6(1/0)	4(0/0)	10(2/0)	4(0/0)
d(An)	Address register indirect with displacement	8(2/0)	4(0/0)	12(3/0)	4(1/0)
d(An, ix)*	Address register indirect with index	10(2/0)	8(1/0)	14(3/0)	8(1/0)
xxx.W	Absolute short	8(2/0)	4(1/0)	12(3/0)	4(1/0)
xxx.L	Absolute long	12(3/0)	8(2/0)	16(4/0)	8(2/0)
d(PC)	Program counter with displacement	8(2/0)	–	12(3/0)	–
d(PC, ix)	Program counter with index	10(2/0)	–	14(3/0)	–
#xxx	Immediate	4(1/0)	–	8(2/0)	–

NOTE:
*The size of the index register (ix) does not affect execution time.

Table 26. Move Byte and Word Instruction Execution Times

SOURCE	DESTINATION								
	Dn	An	(An)	(An)+	−(An)	d(An)	d(An, ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
An	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
(An)	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
(An)+	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
−(An)	10(2/0)	10(2/0)	14(2/1)	14(2/1)	14(2/1)	18(3/1)	20(3/1)	18(3/1)	22(4/1)
d(An)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d(An, ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
xxx.W	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
xxx.L	16(4/0)	16(4/0)	20(4/1)	20(4/1)	20(4/1)	24(5/1)	26(5/1)	24(5/1)	28(6/1)
d(PC)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d(PC, ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
#xxx	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)

NOTE:
*The size of the index register (ix) does not affect execution time.

Move Instruction Execution Times
Tables 26, 27, 28, and 29 indicate the number of clock periods for the move instruction. This data includes instruction fetch, operand reads, and operand writes. The number of bus read and write cycles is shown in parenthesis as (r/w).

Table 27. Move Byte and Word Instruction Loop Mode Execution Times

	LOOP CONTINUED			LOOP TERMINATED					
	Valid Count, cc False			Valid Count, cc True			Expired Count		
	Destination								
SOURCE	(An)	(An)+	−(An)	(An)	(An)+	−(An)	(An)	(An)+	−(An)
Dn	10(0/1)	10(0/1)	–	18(2/1)	18(2/1)	–	16(2/1)	16(2/1)	–
An*	10(0/1)	10(0/1)	–	18(2/1)	18(2/1)	–	16(2/1)	16(2/1)	–
(An)	14(1/1)	14(1/1)	16(1/1)	20(3/1)	20(3/1)	22(3/1)	18(3/1)	18(3/1)	20(3/1)
(An)+	14(1/1)	14(1/1)	16(1/1)	20(3/1)	20(3/1)	22(3/1)	18(3/1)	18(3/1)	20(3/1)
−(An)	16(1/1)	16(1/1)	18(1/1)	22(3/1)	22(3/1)	24(3/1)	20(3/1)	20(3/1)	22(3/1)

NOTE:
*Word only.

Table 28. Move Long Instruction Execution Times

SOURCE	DESTINATION								
	Dn	An	(An)	(An)+	−(An)	d(An)	d(An, ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	12(1/2)	12(1/2)	14(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
An	4(1/0)	4(1/0)	12(1/2)	12(1/2)	14(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
(An)	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
(An)+	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
−(An)	14(3/0)	14(3/0)	22(3/2)	22(3/2)	22(3/2)	26(4/2)	28(4/2)	26(4/2)	30(5/2)
d(An)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
d(An, ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
xxx.W	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
xxx.L	20(5/0)	20(5/0)	28(5/2)	28(5/2)	28(5/2)	32(6/2)	34(6/2)	32(6/2)	36(7/2)
d(PC)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(5/2)
d(PC, ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
#xxx	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)

NOTE:
*The size of the index register (ix) does not affect execution time.

Table 29. Move Long Instruction Loop Mode Execution Times

	LOOP CONTINUED			LOOP TERMINATED					
	Valid Count, cc False			Valid Count, cc True			Expired Count		
	Destination								
SOURCE	**(An)**	**(An)+**	**–(An)**	**(An)**	**(An)+**	**–(An)**	**(An)**	**(An)+**	**–(An)**
Dn	14(0/2)	14(0/2)	–	20(2/2)	20(2/2)	–	18(2/2)	18(2/2)	–
An	14(0/2)	14(0/2)	–	20(2/2)	20(2/2)	–	18(2/2)	18(2/2)	–
(An)	22(2/2)	22(2/2)	24(2/2)	28(4/2)	28(4/2)	30(4/2)	24(4/2)	24(4/2)	26(4/2)
(An)+	22(2/2)	22(2/2)	24(2/2)	28(4/2)	28(4/2)	30(4/2)	24(4/2)	24(4/2)	26(4/2)
–(An)	24(2/2)	24(2/2)	26(2/2)	30(4/2)	30(4/2)	32(4/2)	26(4/2)	26(4/2)	28(4/2)

Standard Instruction Execution Times

The number of clock periods shown in Tables 30 and 31 indicate the time required to perform the operations, store the results, and read the next instruction. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

In Tables 30 and 31 the headings have the following meanings: An = address register operand, Dn = data register operand, ea = an operand specified by an effective address, and M = memory effective address operand.

Table 30. Standard Instruction Execution Times

INSTRUCTION	SIZE	op <ea>, An***	op <ea>, Dn	op Dn, <M>
ADD	byte, word	8(1/0)+	4(1/0)+	8(1/1)+
	long	6(1/0)+	6(1/0)	12(1/2)+
AND	byte, word	–	4(1/0)+	8(1/1)+
	long	–	6(1/0)+	12(1/2)+
CMP	byte, word	6(1/0)+	4(1/0)+	–
	long	6(1/0)+	6(1/0)+	–
DIVS	–	–	122(1/0)+	–
DIVU	–	–	108(1/0)+	–
EOR	byte, word	–	4(1/0)+ **	18(1/1)+
	long	–	6(1/0)**	12(1/2)+
MULS	–	–	42(1/0)+ *	–
MULU	–	–	40(1/0)+	–
OR	byte, word	–	4(1/0)+	8(1/1)+
	long	–	6(1/0)+	12(1/2)+
SUB	byte, word	8(1/0)+	4(1/0)+	8(1/1)+
	long	6(1/0)+	6(1/0)+	12(1/2)+

NOTES:

+Add effective address calculation time

*Indicates maximum value

**Only available addressing mode is data register direct

***Word or long only

Table 31. Standard Instruction Loop Mode Execution Times

		LOOP CONTINUED			LOOP TERMINATED					
		Valid Count, cc False			Valid Count, cc True			Expired Count		
INSTRUCTION	SIZE	op <ea>, An*	op <ea>, Dn	op Dn, <ea>	op <ea>, An*	op <ea>, Dn	op Dn, <ea>	op <ea>, An*	op <ea>, Dn	op Dn, <ea>
ADD	byte, word	18(1/0)	16(1/0)	16(1/1)	24(3/0)	22(3/0)	22(3/1)	22(3/0)	20(3/0)	20(3/1)
	long	22(2/0)	22(2/0)	24(2/2)	28(4/0)	28(4/0)	30(4/2)	26(4/0)	26(4/0)	28(4/2)
AND	byte, word	–	16(1/0)	16(1/1)	–	22(3/0)	22(3/1)	–	20(3/0)	20(3/1)
	long	–	22(2/0)	24(2/2)	–	28(4/0)	30(4/2)	–	26(4/0)	28(4/2)
CMP	byte, word	12(1/0)	12(1/0)	–	18(3/0)	18(3/0)	–	16(3/0)	16(4/0)	–
	long	18(2/0)	18(2/0)	–	24(4/0)	24(4/0)	–	20(4/0)	20(4/0)	–
EOR	byte, word	–	–	16(1/0)	–	–	22(3/1)	–	–	20(3/1)
	long	–	–	24(2/2)	–	–	30(4/2)	–	–	28(4/2)
OR	byte, word	–	16(1/0)	16(1/0)	–	22(3/0)	22(3/1)	–	20(3/0)	20(3/1)
	long	–	22(2/0)	24(2/2)	–	28(4/0)	30(4/2)	–	26(4/0)	28(4/2)
SUB	byte, word	18(1/0)	16(1/0)	16(1/1)	24(3/0)	22(3/0)	22(3/1)	22(3/0)	20(3/0)	20(3/1)
	long	22(2/0)	20(2/0)	24(2/2)	28(4/0)	26(4/0)	(30(4/2)	26(4/0)	24(4/0)	28(4/2)

NOTES:
*Word or long only.
<ea> may be (An), +(An), or −(An) only. Add two clock periods to the table value if <ea> is −(An).

Immediate Instruction Execution Times
The number of clock periods shown in Table 32 includes the time to fetch immediate operands, perform the operations, store the results, and read the next operation. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

In Table 32, the headings have the following meanings: # = immediate operand, Dn = data register operand, AN = address register operand, and M = memory operand.

Table 32. Immediate Instruction Execution Times

INSTRUCTION	SIZE	op #, Dn	op #, An	op #, M
ADDI	byte, word	8(2/0)	–	12(2/1)+
	long	14(3/0)	–	20(3/2)+
ADDQ	byte, word	4(1/0)	4(1/0)*	8(1/1)+
	long	8(1/0)	8(1/0)	12(1/2)+
ANDI	byte, word	8(2/0)	–	12(2/1)+
	long	14(3/0)	–	20(3/1)+
CMPI	byte, word	8(2/0)	–	8(2/0)+
	long	12(3/0)	–	12(3/0)+
EORI	byte, word	8(2/0)	–	12(2/1)+
	long	14(3/0)	–	20(3/2)+
MOVEQ	long	4(1/0)	–	–
ORI	byte, word	8(2/0)	–	12(2/1)+
	long	14(3/0)	–	20(3/2)+
SUBI	byte, word	8(2/0)	–	12(2/1)+
	long	14(3/0)	–	20(3/2)+
SUBQ	byte, word	4(1/0)	4(1/0)*	8(1/1)+
	long	8(1/0)	8(1/0)	12(1/2)+

NOTES:
+ Add effective address calculation time
*Word only

Single Operand Instruction Execution Times

Tables 33, 34, and 35 indicate the number of clock periods for the single operand instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 33. Single Operand Instruction Execution Times

INSTRUCTION	SIZE	REGISTER	MEMORY
NBCD	byte	6(1/0)	8(1/1)+
NEG	byte, word	4(1/0)	8(1/1)+
	long	6(1/0)	12(1/2)+
NEGX	byte, word	4(1/0)	8(1/1)+
	long	6(1/0)	12(1/2)+
NOT	byte, word	4(1/0)	8(1/1)+
	long	6(1/0)	12(1/2)+
S_{CC}	byte, false	4(1/0)	8(1/1)+*
	byte, true	4(1/0)	8(1/1)+*
TAS	byte	4(1/0)	14(2/1)+*
TST	byte, word	4(1/0)	4(1/0)
	long	4(1/0)	4(1/0)+

NOTES:
+ Add effective address calculation time
*Use non-fetching effective address calculation time.

Table 34. Clear Instruction Execution Times

	SIZE	Dn	An	(An)	(An)+	−(An)	d(An)	(An, ix)*	xxx.W	xxx.L
CLR	byte, word	4(1/0)	–	8(1/1)	8(1/1)	10(1/1)	12(2/1)	16(2/1)	12(2/1)	16(3/1)
	long	6(1/0)	–	12(1/2)	12(1/2)	14(1/2)	16(2/2)	20(2/2)	16(2/2)	20(3/2)

NOTE:
*The size of the index register (ix) does not affect execution time.

Table 35. Single Operand Instruction Loop Mode Execution Times

		LOOP CONTINUED			LOOP TERMINATED					
		Valid Count, cc False			Valid Count, cc True			Expired Count		
INSTRUCTION	SIZE	(An)	(An)+	−(An)	(An)	(An)+	−(An)	(An)	(An)+	−(An)
CLR	byte, word	10(0/1)	10(0/1)	12(0/1)	18(2/1)	18(2/1)	20(2/0)	16(2/1)	16(2/1)	18(2/1)
	long	14(0/2)	14(0/2)	16(0/2)	22(2/2)	22(2/2)	24(2/2)	20(2/2)	20(2/2)	22(2/2)
NBCD	byte	18(1/1)	18(1/1)	20(1/1)	24(3/1)	24(3/1)	26(3/1)	22(3/1)	22(3/1)	24(3/1)
NEG	byte, word	16(1/1)	16(1/1)	18(2/2)	22(3/1)	22(3/1)	24(3/1)	20(3/1)	20(3/1)	22(3/1)
	long	24(2/2)	24(2/2)	26(2/2)	30(4/2)	30(4/2)	32(4/2)	28(4/2)	28(4/2)	30(4/2)
NEGX	byte, word	16(1/1)	16(1/1)	18(2/2)	22(3/1)	22(3/1)	24(3/1)	20(3/1)	20(3/1)	22(3/1)
	long	24(2/2)	24(2/2)	26(2/2)	30(4/2)	30(4/2)	32(4/2)	28(4/2)	28(4/2)	30(4/2)
NOT	byte, word	16(1/1)	16(1/1)	18(2/2)	22(3/1)	22(3/1)	24(3/1)	20(3/1)	20(3/1)	22(3/1)
	long	24(2/2)	24(2/2)	26(2/2)	30(4/2)	30(4/2)	32(4/2)	28(4/2)	28(4/2)	30(4/2)
TST	byte, word	12(1/0)	12(1/0)	14(1/0)	18(3/0)	18(3/0)	20(3/0)	16(3/0)	16(3/0)	18(3/0)
	long	18(2/0)	18(2/0)	20(2/0)	24(4/0)	24(4/0)	26(4/0)	20(4/0)	20(4/0)	22(4/0)

Shift/Rotate Instruction Execution Times

Tables 36 and 37 indicate the number of clock periods for the shift and rotate instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 36. Shift/Rotate Instruction Execution Times

INSTRUCTION	SIZE	REGISTER	MEMORY*
ASR, ASL	byte, word	6 + 2n(1/0)	8(1/1)+
	long	8 + 2n(1/0)	–
LSR, LSL	byte, word	6 + 2n(1/0)	8(1/1)+
	long	8 + 2n(1/0)	–
ROR, ROL	byte, word	6 + 2n(1/0)	8(1/1)+
	long	8 + 2n(1/0)	–
ROXR, ROXL	byte, word	6 + 2n(1/0)	8(1/1)+
	long	8 + 2n(1/0)	–

NOTES:
+ Add effective address calculation time
n Is the shift or rotate count
*Word only

Table 37. Shift/Rotate Instruction Loop Mode Execution Times

		LOOP CONTINUED			LOOP TERMINATED					
		Valid Count, cc False			Valid Count, cc True			Expired Count		
INSTRUCTION	SIZE	(An)	(An)+	–(An)	(An)	(An)+	–(An)	(An)	(An)+	–(An)
ASR, ASL	word	18(1/1)	18(1/1)	20(1/1)	24(3/1)	24(3/1)	26(3/1)	22(3/1)	22(3/1)	24(3/1)
LSR, LSL	word	18(1/1)	18(1/1)	20(1/1)	24(3/1)	24(3/1)	26(3/1)	22(3/1)	22(3/1)	24(3/1)
ROR, ROL	word	18(1/1)	18(1/1)	20(1/1)	24(3/1)	24(3/1)	26(3/1)	22(3/1)	22(3/1)	24(3/1)
ROXR, ROXL	word	18(1/1)	18(1/1)	20(1/1)	24(3/1)	24(3/1)	26(3/1)	22(3/1)	22(3/1)	24(3/1)

Bit Manipulation Instruction Execution Times

Table 38 indicates the number of clock periods required for the bit manipulation instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

Table 38. Bit Manipulation Instruction Execution Times

INSTRUCTION	SIZE	DYNAMIC		STATIC	
		Register	Memory	Register	Memory
BCHG	byte	–	8(1/1)+	–	12(2/1)+
	long	8(1/0)*	–	12(2/0)*	–
BCLR	byte	–	10(1/1)+	–	14(2/1)+
	long	10(1/0)*	–	14(2/0)*	–
BSET	byte	–	8(1/0)+	–	12(2/1)+
	long	8(1/0)*	–	12(2/0)*	–
BTST	byte	–	4(1/0)+	–	8(2/0)+
	long	6(1/0)*	–	10(2/0)	–

NOTES:
+ Add effective address calculation time
*Indicates maximum value

Conditional Instruction Execution Times
Table 39 indicates the number of clock periods required for the conditional instructions. The number of bus read and write cycles is indicated in parenthesis as (r/w). The number of clock periods and the number of read and write cycles must be added respectively to those of the effective address calculation where indicated.

JMP, JSR, LEA, PEA, and MOVEM Instruction Execution Times
Table 40 indicates the number of clock periods required for the jump, jump-to-subroutine, load effective address, push effective address, and move multiple registers instructions. The number of bus read and write cycles is shown in parenthesis as (r/w).

Table 39. Conditional Instruction Execution Times

INSTRUCTION	DISPLACEMENT	BRANCH TAKEN	BRANCH NOT TAKEN
B_{CC}	byte	10(2/0)	6(1/0)
	word	10(2/0)	10(2/0)
BRA	byte	10(2/0)	–
	word	10(2/0)	–
BSR	byte	18(2/2)	–
	word	18(2/2)	–
DB_{CC}	cc true	–	10(2/0)
	cc false	10(2/0)	16(3/0)

Table 40. JMP, JSR, LEA, PEA, and Movem Instruction Execution Times

INSTRUCTION	SIZE	(An)	(An)+	–(An)	d(An)	d(An, ix)+	xxx.W	xxx.L	d(PC)	d(PC, ix)*
JMP	–	8(2/0)	–	–	10(2/0)	14(3/0)	10(2/0)	12(3/0)	10(2/0)	14(3/0)
JSR	–	16(2/2)	–	–	18(2/2)	22(2/2)	18(2/2)	20(3/2)	18(2/2)	22(2/2)
LEA	–	4(1/0)	–	–	8(2/0)	12(2/0)	8(2/0)	12(3/0)	8(2/0)	12(2/0)
PEA	–	12(1/2)	–	–	16(2/2)	20(2/2)	16(2/2)	20(3/2)	16(2/2)	20(2/2)
MOVEM M → R	word	12 + 4n (3 + n/0)	12 + 4n (3 + n/0)	–	16 + 4n (4 + n/0)	18 + 4n (4 + n/0)	16 + 4n (4 + n/0)	20 + 4n (5 + n/0)	16 + 4n (4 + n/0)	18 + 4n (4 + n/0)
	long	12 + 8n (3 + 2n/0)	12 + 8n (3 + 2n/0)	–	16 + 8n (4 + 2n/0)	18 + 8n (4 + 2n/0)	16 + 8n (4 + 2n/0)	20 + 8n (5 + 2n/0)	16 + 8n (4 + 2n/0)	18 + 8n (4 + 2n/0)
MOVEM R → M	word	8 + 4n (2/n)	–	8 + 4n (2/n)	12 + 4n (3/n)	14 + 4n (3/n)	12 + 4n (3/n)	16 + 4n (4/n)	– –	– –
	long	8 + 8n (2/2n)	– –	8 + 8n (2/2n)	12 + 8n (3/2n)	14 + 8n (3/2n)	12 + 8n (3/2n)	16 + 8n (4/2n)	– –	– –

NOTES:
n Is the number of registers to move
*The size of the index register (ix) does not affect the instruction's execution time

Multi-Precision Instruction Execution Times
Table 41 indicates the number of clock periods for the multi-precision instructions. The number of clock periods includes the time to fetch both operands, perform the operations, store the results, and read the next instructions. The number of read and write cycles is shown in parenthesis as (r/w).

In Table 41, the headings have the following meanings: Dn = data register operand and M = memory operand.

Table 41. Multi-Precision Instruction Execution Times

		NON-LOOPED		LOOP MODE		
				Continued	Terminated	
				Valid Count, cc False	Valid Count, ccTrue	Expired Count
INSTRUCTION	SIZE	op Dn, Dn	op M, M*			
ADDX	byte, word	4(1/0)	18(3/10)	22(2/1)	28(4/1)	26(4/1)
	long	6(1/0)	30(5/2)	32(4/2)	38(6/2)	36(6/2)
CMPM	byte, word	–	12(3/0)	14(2/0)	20(4/0)	18(4/0)
	long	–	20(5/0)	24(4/0)	30(6/0)	26(6/0)
SUBX	byte, word	4(1/0)	18(3/1)	22(2/1)	28(4/1)	26(4/1)
	long	6(1/0)	30(5/2)	32(4/2)	38(6/2)	36(6/2)
ABCD	byte	6(1/0)	18(3/1)	24(2/1)	30(4/1)	28(4/1)
SBCD	byte	6(1/0)	18(3/1)	24(2/1)	30(4/1)	28(4/1)

NOTE:
*Source and destination ea is (An)+ for CMPM and –(An) for all others.

Miscellaneous Instruction Execution Times

Table 42 indicates the number of clock periods for the following miscellaneous instructions. The number of bus read and write cycle is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Exception Processing Execution Times

Table 43 indicates the number of clock periods for exception processing. The number of clock periods includes the time for all stacking, the vector fetch, and the fetch of the first two instruction words of the handler routine. The number of bus read and write cycles is shown in parenthesis as (r/w).

Table 43. Exception Processing Execution Times

EXCEPTION	
Address error	126(4/26)
Breakpoint instruction*	42(5/4)
Bus error	126(4/26)
CHK instruction**	44(5/4)+
Divide by zero	42(5/4)
Illegal instruction	38(4/4)
Interrupt*	46(5/4)
MOVEC, illegal cr**	46(5/4)
Privilege violation	38(4/4)
Reset***	40(6/0)
RTE, illegal format	50(7/4)
RTE, illegal revision	70(12/4)
Trace	38(4/4)
TRAP instruction	38(4/4)
TRAPV instruction	40(5/4)

NOTES:

+ Add effective address calculation time.
*The interrupt acknowledge and breakpoint cycles are assumed to take four clock periods.
**Indicates maximum value
***Indicates the time from when $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ are first sampled as negated to when instruction execution starts.

Table 42. Miscellaneous Instruction Execution Times

INSTRUCTION	SIZE	REGISTER	MEMORY	REGISTER → DESTINATION**	SOURCE** → REGISTER
ANDI to CCR	–	16(2/0)	–	–	–
ANDI to SR	–	16(2/0)	–	–	–
CHK	–	8(1/0)+	–	–	–
EORI to CCR	–	16(2/0)	–	–	–
EORI to SR	–	16(2/0)	–	–	–
EXG	–	6(1/0)	–	–	–
EXT	word	4(1/0)	–	–	–
	long	4(1/0)	–	–	–
LINK	–	16(2/2)	–	–	–
MOVE from CCR	–	4(1/0)	8(1/1)+*	–	–
MOVE to CCR	–	12(2/0)	12(2/0)+	–	–
MOVE from SR	–	4(1/0)	8(1/1)+*	–	–
MOVE to SR	–	12(2/0)	12(2/0)+	–	–
MOVE from USP	–	6(1/0)	–	–	–
MOVE to USP	–	6(1/0)	–	–	–
MOVEC	–	–	–	10(2/0)	12(2/0)
MOVEP	word	–	–	16(2/2)	16(4/0)
	long	–	–	24(2/4)	24(6/0)
MOVES	byte, word	–	–	16(2/1) + *	16(3/0) + *
	long	–	–	20(2/2) + *	20(4/0) + *
NOP	–	4(1/0)	–	–	–
ORI to CCR	–	16(2/0)	–	–	–
ORI to SR	–	16(2/0)	–	–	–
RESET	–	130(1/0)	–	–	–
RTD	–	16(4/0)	–	–	–
RTE	short	24(6/0)	–	–	–
	long, retry read	112(27/10)	–	–	–
	long, retry write	112(26/1)	–	–	–
	long, no retry	110(26/0)	–	–	–
RTR	–	20(5/0)	–	–	–
RTS	–	16(4/0)	–	–	–
STOP	–	4(0/0)	–	–	–
SWAP	–	4(1/0)	–	–	–
TRAPV	–	4(1/0)	–	–	–
UNLK	–	12(3/0)	–	–	–

NOTES:

+ Add effective address calculation time.
* Use non-fetching effective address calculation time.
** Source or destination is a memory location for the MOVEP and MOVES instruction and a control register for the MOVEC instruction.

ABSOLUTE MAXIMUM RATINGS

SYMBOL	RATING	VALUE	UNIT
V_{CC}	Supply voltage range	−0.3 to +7.0	V
V_{in}	Input voltage range	−0.3 to +7.0	V
T_A	Operating temperature range SCN68010 SCN68010 ceramic	T_L to T_H 0 to 70 −40 to 85	°C
T_{stg}	Storage temperature range	−55 to 150	°C

NOTE:

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

CHARACTERISTIC	SYMBOL	VALUE	SYMBOL	VALUE	RATING
Thermal Resistance (Still Air)	θ_{JA}		θ_{JC}		°C/W
Ceramic DIP		30		15*	
Pin grid array		33		15	
Plastic DIP		30		15*	
PLCC		45*		25*	

NOTE:

*Estimated

Table 44. Maximum Power Dissipation by Package Type Modes

PACKAGE TYPE	TEMPERATURE (°C)	MAXIMUM POWER DISSIPATION (WATTS) PER FREQUENCY (MHz)		
		8 MHz	10 MHz	12.5 MHz
Ceramic DIP	0 to 70 −40 to 85	1.50 1.65	1.50 1.65	1.75 –
Plastic DIP	0 to 70 −40 to 85	1.50 1.65	1.50 1.65	– –
PGA	0 to 70 −40 to 85	1.50 1.65	1.50 1.65	1.75 –
PLCC	0 to 70	1.50	1.50	–

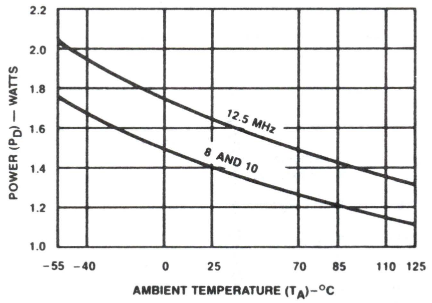


Figure 48. SCN68010 Power Dissipation (P_D) vs Ambient Temperature (T_A)

Power Considerations

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = ambient temperature, °C

θ_{JA} = package thermal resistance, junction-to-ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{CC} \times V_{CC}$, watts — chip internal power

$P_{I/O}$ = power dissipation on input and output pins — user determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = T_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

The curve shown in Figure 48 gives the graphic solution to these equations for the specification power dissipation of 1.50 and 1.75 watts over the ambient temperature range of −55°C to 125°C using a θ_{JA} of 45°C/W for the ceramic package.

The total thermal resistance of a package (θ_{JA}) can be separated into two components, θ_{JC} and θ_{CA}, representing the barrier to heat flow from the semiconductor junction to the package (case) surface (θ_{JC}) and from the case to the outside ambient (θ_{CA}). These terms are related by the equation:

$$\theta_{JA} = \theta_{JC} + \theta_{CA} \quad (4)$$

θ_{JC} is device related and cannot be influenced by the user. However, θ_{CA} is user dependent and can be minimized by such thermal management techniques as heat sinks, ambient air cooling and thermal convention. Thus, good thermal management on the part of the user can significantly reduce θ_{CA} so that θ_{JA} approximately equals θ_{JC}. Substitution of θ_{JC} for θ_{JA} in equation (1) will result in a lower semiconductor junction temperature.

Values for thermal resistance presented in this data sheet, unless estimated, are provided for design purposes only. Thermal measurements are complex and dependent on procedure and setup. User derived values for thermal resistance may differ.

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V_{DC} \pm 5\%$; GND = $0V_{DC}$; $T_A = T_L$ to T_H (see Figures 49, 50, and 51)

SYMBOL	CHARACTERISTIC		MIN	MAX	UNIT
V_{IH}	Input high voltage		2	V_{CC}	V
V_{IL}	Input low voltage		GND – 0.3	0.8	V
I_{IN}	Input leakage current @ 5.25V	BERR, BGACK, BR, DTACK, CLK, IPL0 – IPL2, VPA HALT, RESET	–	2.5 20	μA
I_{TSI}	3-State (off state) input current @ 2.4V/0.4V	$\overline{AS}$, A1 – A23, D0 – D15, FC0 – FC2, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$	–	20	μA
V_{OH}	Output high voltage ($I_{OH} = -400\mu A$)	E*	$V_{CC} - 0.75$	–	V
		E**, $\overline{AS}$, A1 – A23, $\overline{BG}$, D0 – D15, FC0 – FC2, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$	2.4	–	
V_{OL}	Output low voltage				
	($I_{OL} = 1.6mA$)	$\overline{HALT}$	–	0.5	
	($I_{OL} = 3.2mA$)	A1 – A23, $\overline{BG}$, FC0 – FC2	–	0.5	V
	($I_{OL} = 5.0mA$)	$\overline{RESET}$	–	0.5	
	($I_{OL} = 5.3mA$)	E, $\overline{AS}$, D0 – D7, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$		–	0.5
	Power dissipation (see Power Considerations)***		P_D	–	–
C_{in}	Capacitance ($V_{IN} = 0V$, $T_A = 25°C$, frequency = 1MHz)****		–	20	pF

NOTES:

* With external pullup resistor of 1.1 kΩ.

** Without external pullup resistor.

*** During normal operation instantaneous V_{CC} current requirements may be as high as 1.5A.

****Capacitance is periodically sampled rather than 100% tested.

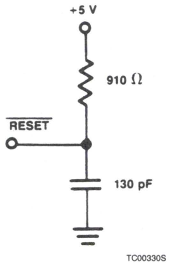


Figure 49. $\overline{RESET}$ Test Load

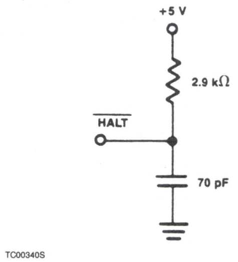
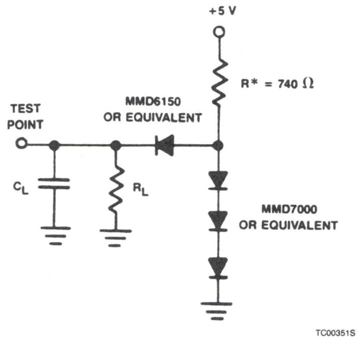


Figure 50. Test Loads



C_L = 130 pF
(INCLUDES ALL PARASITICS)
R_L = 6.0 kΩ FOR
$\overline{AS}$, A1-A23, $\overline{BG}$, D0-D15, E
FC0-FC2, $\overline{LDS}$, R/$\overline{W}$, $\overline{UDS}$, $\overline{VMA}$
*R = 1.22 kΩ FOR A1-A23, $\overline{BG}$,
FC0-FC2

Figure 51. $\overline{HALT}$ Test Load

AC ELECTRICAL CHARACTERISTICS — Clock Input (See Figure 52)

CHARACTERISTIC	SYMBOL	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
Frequency of operation	f	4	8	4	10	4	12.5	MHz
Cycle time	t_{cyc}	125	250	100	250	80	250	ns
Clock pulse width	t_{CL} t_{CH}	55 55	125 125	45 45	125 125	35 35	125 125	ns
Rise and fall times	t_{Cr} t_{Cf}	– –	10 10	– –	10 10		5 5	ns

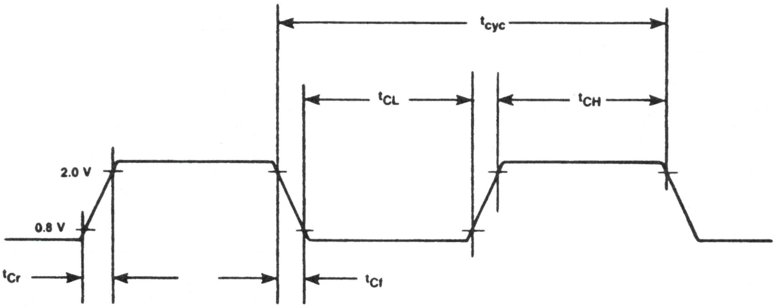


Figure 52. Clock Input Timing Diagram

AC ELECTRICAL CHARACTERISTICS — Read and Write Cycles ($V_{CC} = 5V_{DC} \pm 5\%$; $GND = 0V_{DC}$; $T_A = T_L$ to T_H see Figures 53 and 54)

NO.	CHARACTERISTIC	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
1	Clock period	125	500	100	250	80	250	ns
2	Clock width low	55	125	45	125	35	125	ns
3	Clock width high	55	125	45	125	35	125	ns
4	Clock fall time	–	10	–	10	–	5	ns
5	Clock rise time	–	10	–	10	–	5	ns
6	Clock low to address valid	–	70	–	55	–	55	ns
6A	Clock high to FC valid	–	70	–	55	–	55	ns
7	Clock high to address, data bus high impedance (maximum)	–	80	–	70	–	60	ns
8	Clock high to address, FC invalid (minimum)	0	–	0	–	0	–	ns
9[1]	Clock high to $\overline{\text{AS}}$, $\overline{\text{DS}}$ low	0	60	0	55	0	55	ns
11[2]	Address valid to $\overline{\text{AS}}$, $\overline{\text{DS}}$ low (read)/$\overline{\text{AS}}$ low (write)	30	–	20	–	0	–	ns
11A[2]	FC valid to $\overline{\text{AS}}$, $\overline{\text{DS}}$ low (read)/$\overline{\text{AS}}$ low (write)	60	–	50	–	40	–	ns
12[1]	Clock low to $\overline{\text{AS}}$, $\overline{\text{DS}}$ high	–	70	–	55	–	50	ns
13[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to address/FC invalid	30	–	20	–	10	–	ns
14[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ width low (read)/$\overline{\text{AS}}$ low (write)	240	–	195	–	160	–	ns
14A[2]	$\overline{\text{DS}}$ width low (write)	–	115	95	–	80	–	ns
15[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ width high	150	–	105	–	65	–	ns
16	Clock high to control bus high impedance	–	80	–	70	–	60	ns
17[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to R/$\overline{\text{W}}$ high (read)	40	–	20	–	10	–	ns
18[1]	Clock high to R/$\overline{\text{W}}$ high	0	70	0	60	0	60	ns
20[1]	Clock high to R/$\overline{\text{W}}$ low	–	70	–	60	–	60	ns
20A[2]	$\overline{\text{AS}}$ low to R/$\overline{\text{W}}$ valid (write)	–	20	–	20	–	20	ns
21[2]	Address valid to R/$\overline{\text{W}}$ low (write)	20	–	0	–	0	–	ns
21A[2]	FC valid to R/$\overline{\text{W}}$ low (write)	60	–	50	–	30	–	ns
22[2]	R/$\overline{\text{W}}$ low to $\overline{\text{DS}}$ low (write)	80	–	50	–	30	–	ns
23	Clock low to data out valid (write)	–	70	–	55	–	55	ns
25[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to data out invalid (write)	30	–	20	–	15	–	ns
26[2]	Data out valid to $\overline{\text{DS}}$ low (write)	30	–	20	–	15	–	ns
27[5]	Data in to clock low (setup time on read)	15	–	10	–	10	–	ns
27A	Late $\overline{\text{BERR}}$ low to clock low (setup time)	45	–	45	–	45	–	ns
28[2]	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to $\overline{\text{DTACK}}$ high	0	245	0	190	0	150	ns
29	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to data invalid (hold time on read)	0	–	0	–	0	–	ns
30	$\overline{\text{AS}}$, $\overline{\text{DS}}$ high to $\overline{\text{BERR}}$ high	0	–	0	–	0	–	ns
31[2,5]	$\overline{\text{DTACK}}$ low to data valid (setup time)	–	90	–	65	–	50	ns
32	$\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ input transition time	0	200	0	200	0	200	ns
33	Clock high to $\overline{\text{BG}}$ low	–	70	–	60	–	50	ns
34	Clock high to $\overline{\text{BG}}$ high	–	70	–	60	–	50	ns
35	$\overline{\text{BR}}$ low to $\overline{\text{BG}}$ low	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	clk. per.

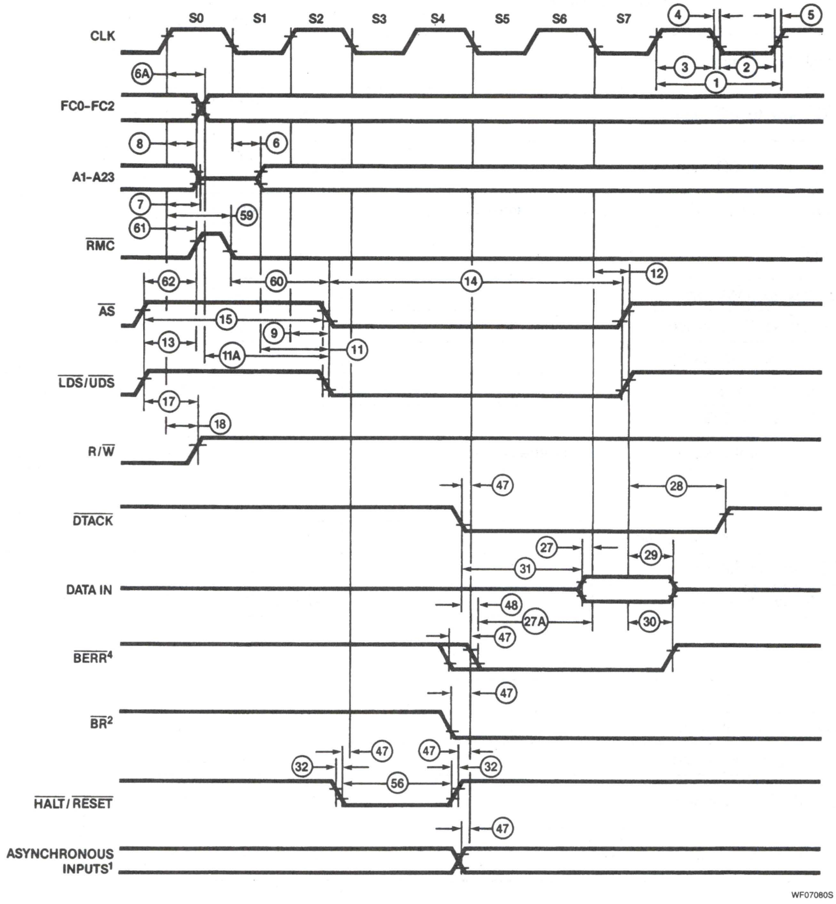
AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	CHARACTERISTIC	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
36[6]	$\overline{\mathrm{BR}}$ high to $\overline{\mathrm{BG}}$ high	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	clk. per.
37	$\overline{\mathrm{BGACK}}$ low to $\overline{\mathrm{BG}}$ high	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	clk. per.
37A[7]	$\overline{\mathrm{BGACK}}$ low to $\overline{\mathrm{BR}}$ high	20	1.5 clocks	20	1.5 clocks	20	1.5 clocks	ns
38	$\overline{\mathrm{BG}}$ low to control, address, data bus high impedance ($\overline{\mathrm{AS}}$ high)	–	80	–	70	–	60	ns
39	$\overline{\mathrm{BG}}$ width high	1.5	–	1.5	–	1.5	–	clk. per.
40	Clock low to $\overline{\mathrm{VMA}}$ low	–	70	–	70	–	70	ns
41	Clock low to E transition	–	70	–	55	–	45	ns
42	E output rise and fall time	–	25	–	25	–	25	ns
43	$\overline{\mathrm{VMA}}$ low to E high	200	–	150	–	90	–	ns
44	$\overline{\mathrm{AS}}$, $\overline{\mathrm{DS}}$ high to $\overline{\mathrm{VPA}}$ high	0	120	0	90	0	70	ns
45	E low to control address bus invalid (address hold time)	30	–	10	–	10	–	ns
46	$\overline{\mathrm{BGACK}}$ width	1.5	–	1.5	–	1.5	–	clk. per.
47[5]	Asynchronous input setup time	20	–	20	–	20	–	ns
48[2,3]	$\overline{\mathrm{DTACK}}$ low to $\overline{\mathrm{BERR}}$ low	–	80	–	55	–	35	ns
49[8]	$\overline{\mathrm{AS}}$, $\overline{\mathrm{DS}}$ high to E low	−70	70	−55	55	−45	45	ns
50	E width high	450	–	350	–	280	–	ns
51	E width low	700	–	550	–	440	–	ns
53	Clock high to data out valid	0	–	0	–	0	–	ns
54	E low to data out valid	30	–	20	–	15	–	ns
55	R/$\overline{\mathrm{W}}$ to data bus driven	30	–	20	–	10	–	ns
56[4]	$\overline{\mathrm{HALT}}$/$\overline{\mathrm{RESET}}$ pulse width	10	–	10	–	10	–	clk. per.
57	$\overline{\mathrm{BGACK}}$ high to control bus driven	1.5	–	1.5	–	1.5	–	clk. per
58[6]	$\overline{\mathrm{BG}}$ high to control bus driven	1.5	–	1.5	–	1.5	–	clk. per

NOTES:

1. For a loading capacitance of less than or equal to 50pF, subtract 5ns from the values given in these columns.
2. Actual value depends on clock period.
3. In the absence of $\overline{\mathrm{DTACK}}$, $\overline{\mathrm{BERR}}$ is an asynchronous input using the asynchronous input setup time (#47).
4. For power-up, the MPU must be held in $\overline{\mathrm{RESET}}$ state for 100ms to allow stabilization of on-chip circuitry. After the system is powered up, #56 refers to the minimum pulse width required to reset the system.
5. If the asynchronous setup time (#47) requirements are satisfied, the $\overline{\mathrm{DTACK}}$-low to data setup time (#31) and $\overline{\mathrm{DTACK}}$–low to $\overline{\mathrm{BERR}}$–low setup time (#48) requirements can be ignored. The data must only satisfy the data-in to clock-low setup time (#27) for the following clock cycle, $\overline{\mathrm{BERR}}$ must only satisfy the late-$\overline{\mathrm{BERR}}$-low to clock-low setup time (#27A) for the following clock cycle.
6. The processor will negate $\overline{\mathrm{BG}}$ and begin driving the bus again if external arbitration logic negates $\overline{\mathrm{BR}}$ before asserting $\overline{\mathrm{BGACK}}$.
7. The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{\mathrm{BG}}$ may be reasserted.
8. The falling edge of S6 triggers both the negation of the strobes ($\overline{\mathrm{AS}}$ and $\overline{\mathrm{DS}}$) and the falling edge of E. Either of these events can occur first, depending on the loading on each signal. Specification #49 indicates the absolute maximum skew that will occur between the rising edge of the strobes and the falling edge of the E clock.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

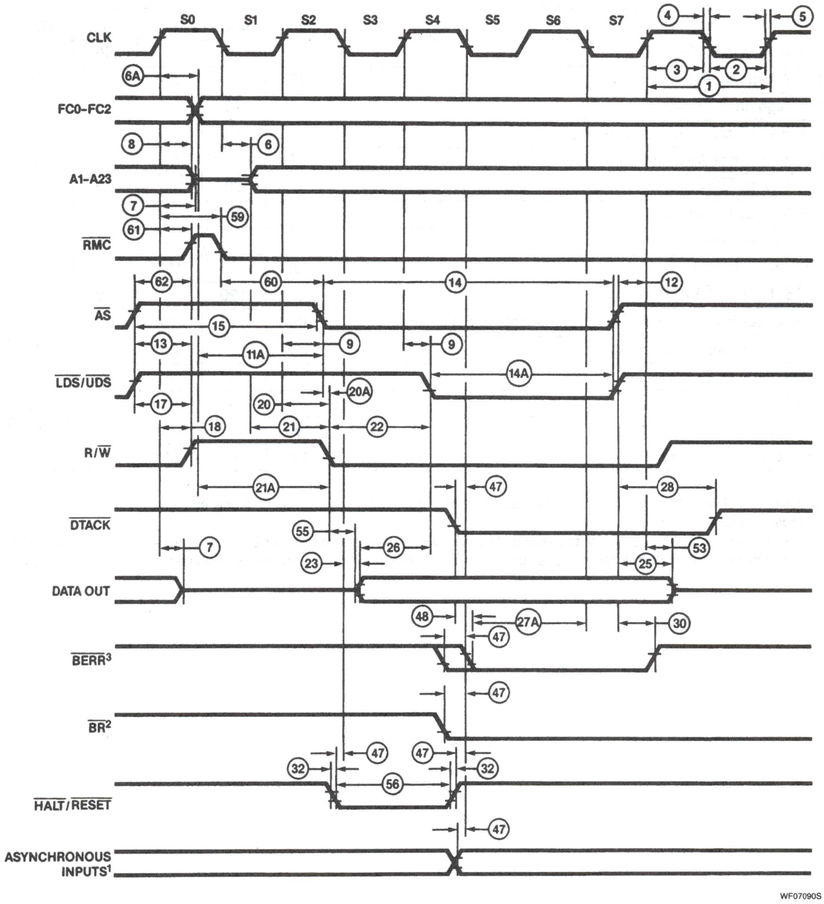

S0
S1
S2
S3
S4
S5
S6
S7
CLK
④
⑤
③
②
①
6A
FC0-FC2
⑧
⑥
A1-A23
⑦
61
59
RMC
62
60
14
12
AS
15
⑨
13
11A
11
LDS/UDS
17
18
R/W
47
28
DTACK
27
29
31
DATA IN
48
27A
30
47
BERR⁴
47
BR²
32
47
47
32
56
HALT/RESET
47
ASYNCHRONOUS INPUTS¹
WF07080S

NOTES:

1. Setup time for the asynchronous inputs IPL0/2, IPL1, and VPA guarantees their recognition at the next falling edge of the clock.
2. BR need fall at this time only in order to insure being recognized at the end of this bus cycle.
3. Timing measurements are referenced to and from a low voltage of 0.8V and a high voltage of 2.0V, unless otherwise noted.
4. The timing for the first falling edge (47) of BERR are for BERR without DTACK, the timings for the second falling edge (27A and 48) are for BERR with DTACK.

Figure 53. Read Cycle Timing Diagram

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.


S0
S1
S2
S3
S4
S5
S6
S7
CLK
④
⑤
③
②
①
6A
FC0-FC2
⑧
⑥
A1-A23
⑦
(59)
(61)
$\overline{RMC}$
(62)
(60)
⑭
⑫
$\overline{AS}$
⑮
⑬
11A
⑨
⑨
14A
$\overline{LDS}/\overline{UDS}$
20A
⑰
⑳
⑱
㉑
㉒
$R/\overline{W}$
21A
(47)
㉘
$\overline{DTACK}$
(55)
⑦
㉖
(53)
㉓
㉕
DATA OUT
(48)
27A
㉚
(47)
$\overline{BERR}^3$
(47)
$\overline{BR}^2$
(47)
(47)
(32)
(32)
(56)
$\overline{HALT}/\overline{RESET}$
(47)
ASYNCHRONOUS INPUTS^1
WF07090S

NOTES:

1. Timing measurements are referenced to and from a low voltage of 0.8V and a high voltage of 2.0V, unless otherwise noted.
2. Because of loading variations, $R/\overline{W}$ may be valid after $\overline{AS}$ even though both are initiated by the rising edge of S2 (specification 20A).
3. The timing for the first falling edge (47) of $\overline{BERR}$ are for $\overline{BERR}$ without $\overline{DTACK}$; the timings for the second falling edge (27A and 48) are for $\overline{BERR}$ with $\overline{DTACK}$.

Figure 54. Write Cycle Timing Diagram

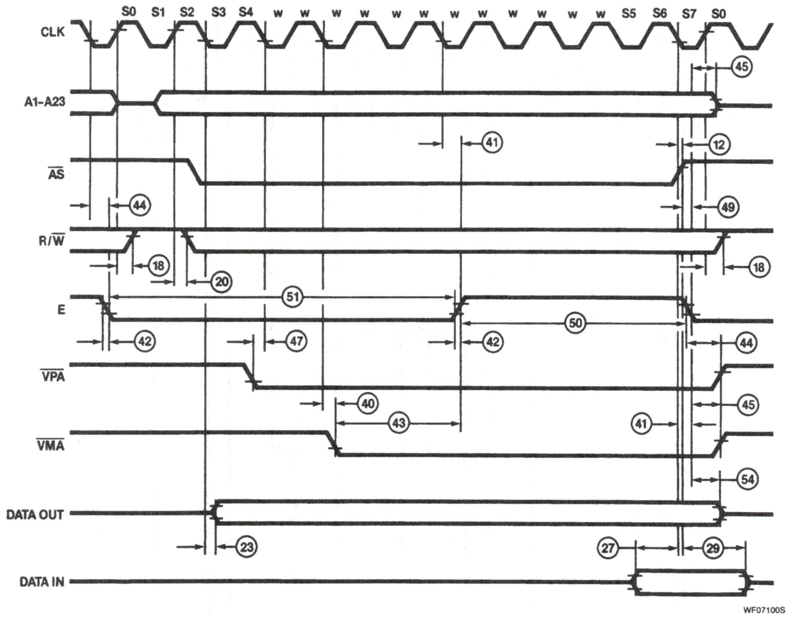
AC ELECTRICAL CHARACTERISTICS — SCN68010 to Synchronous Peripheral Cycles $V_{CC} = 5V_{DC}$ ±5%, GND = $0V_{DC}$, $T_A = T_L$ to T_H (refer to Figures 55 and 56)

NO.	CHARACTERISTIC	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
12[1]	Clock low to $\overline{AS}$, $\overline{DS}$ high	–	70	–	55	–	50	ns
17[2]	$\overline{AS}$, $\overline{DS}$ high to R/$\overline{W}$ high (read)	40	–	20	–	10	–	ns
20[1]	Clock high to R/$\overline{W}$ low	–	70	–	60	–	60	ns
23	Clock low to data out valid (write)	–	70	–	55	–	55	ns
27	Data in to clock low (setup time on read)	15	–	10	–	10	–	ns
40	Clock low to $\overline{VMA}$ low	–	70	–	70	–	70	ns
41	Clock low to E transition	–	70	–	55	–	45	ns
42	E output rise and fall time	–	25	–	25	–	25	ns
43	$\overline{VMA}$ low to E high	200	–	150	–	90	–	ns
44	$\overline{AS}$, $\overline{DS}$ high to $\overline{VPA}$ high	0	120	0	90	0	70	ns
45	E low to control address bus (address hold time)	30	–	10	–	10	–	ns
47	Asynchronous input setup time	20	–	20	–	20	–	ns
49[3]	$\overline{AS}$, $\overline{DS}$ high to E low	–70	70	–55	55	–45	45	ns
50	E width high	450	–	350	–	280	–	ns
51	E width low	700	–	550	–	440	–	ns
54	E low to data out invalid	30	–	20	–	15	–	ns

NOTES:

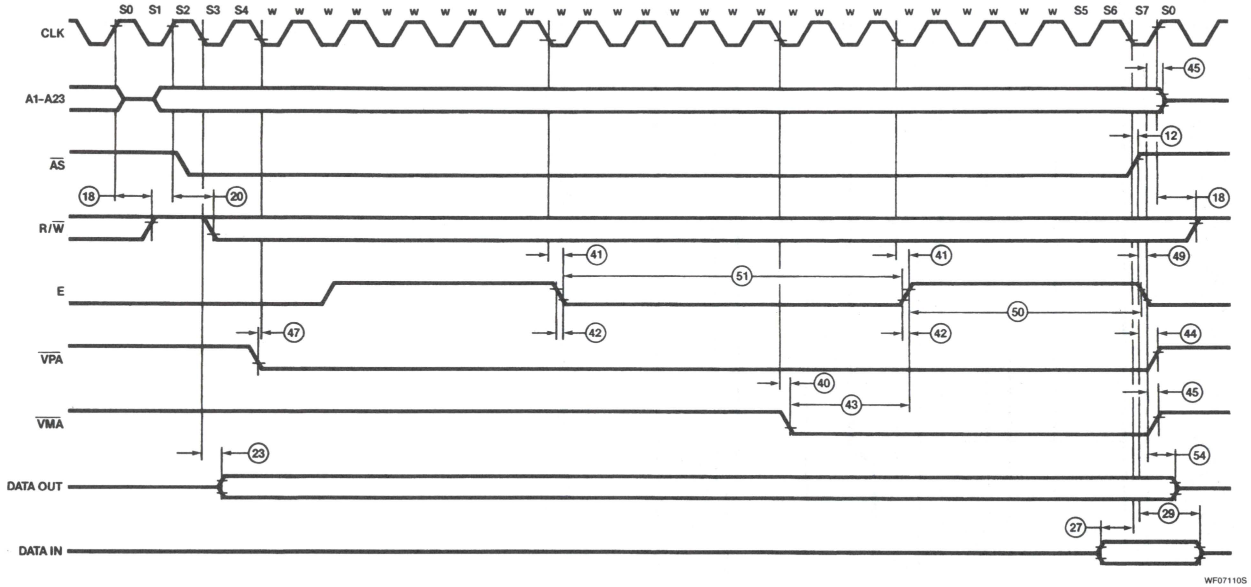
1. For a loading capacitance of less than or equal to 50pF, subtract 5ns from the values given in these columns.
2. Actual value depends on clock period.
3. The falling edge of S6 triggers both the negation of the strobes ($\overline{AS}$ and $\overline{DS}$) and the falling edge of E. Either of these events can occur first, depending on the loading on each signal. Specification #49 indicates the absolute maximum skew that will occur between the rising edge of the strobes and the falling edge of the E clock.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.


S0 S1 S2 S3 S4 w w w w w w w w w w w w S5 S6 S7 S0
CLK
A1-A23
AS
R/W
E
VPA
VMA
DATA OUT
DATA IN
45
41
12
44
49
18
20
18
51
50
42
47
42
44
40
43
41
45
54
23
27
29
WF07100S

NOTE:
This timing diagram is included for those who wish to design their own timing circuit to generate $\overline{\text{VMA}}$. It shows the best case possibly attainable.

Figure 55. SCN68010 to Synchronous Peripheral Timing Diagram — Best Case


S0
S1
S2
S3
S4
w
S5
S6
S7
S0
CLK
A1-A23
AS
R/W
E
VPA
VMA
DATA OUT
DATA IN
18
20
45
12
18
41
41
49
51
50
47
42
42
44
40
43
45
23
54
29
27
WF07110S

NOTE:
This timing diagram is included for those who wish to design their own circuit to generate VMA. It shows the worst case possibly attainable.

Figure 56. SCN68010 to Synchronous Peripheral Timing Diagram — Worst Case

AC ELECTRICAL CHARACTERISTICS — Bus Arbitration $V_{CC}=5V_{DC}$ ±5%; GND = $0V_{DC}$; $T_A = T_L$ to T_H (see Figures 57, 58, 59)

NO.	CHARACTERISTIC	8MHz		10MHz		12.5MHz		UNIT
		Min	Max	Min	Max	Min	Max	
7	Clock high to address, data bus high impedance (maximum)	–	80	–	70	–	60	ns
16	Clock high to control bus high impedance	–	80	–	70	–	60	ns
33	Clock high to $\overline{BG}$ low	–	70	–	60	–	50	ns
34	Clock high to $\overline{BG}$ high	–	70	–	60	–	50	ns
35	$\overline{BR}$ low to $\overline{BG}$ low	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	clk. per.
36[2]	$\overline{BR}$ high to $\overline{BG}$ high	1.5	90ns +3.5	1.5	80ns +3.5	1.5	70ns +3.5	clk. per.
37	$\overline{BGACK}$ low to $\overline{BG}$ high	1.5	90ns 3.5	1.5	80ns 3.5	1.5	70ns 3.5	clk. per.
37A[3]	$\overline{BGACK}$ low to $\overline{BR}$ high	20	1.5 Clocks	20	1.5 Clocks	20	1.5 Clocks	ns
38	$\overline{BG}$ low to control, address, data bus high impedance ($\overline{AS}$ high)	–	80	–	70	–	60	ns
39	$\overline{BG}$ width high	1.5	–	1.5	–	1.5	–	clk. per
46	$\overline{BGACK}$ width	1.5	–	1.5	–	1.5	–	clk. per.
47	Asynchronous input setup time	20	–	20	–	20	–	ns
57[2]	$\overline{BGACK}$ high to control bus driven	1.5	–	1.5	–	1.5	–	clk. per.
58[1,2]	$\overline{BG}$ high to control bus driven	1.5	–	1.5	–	1.5	–	clk. per.

NOTES:

1. The nanosecond value shown in the specification is the asynchronous input setup time (spec. #47).
2. The processor will negate $\overline{BG}$ and begin driving the bus again if external arbitration logic negates $\overline{BR}$ before asserting $\overline{BGACK}$.
3. The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{BG}$ may be reasserted.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

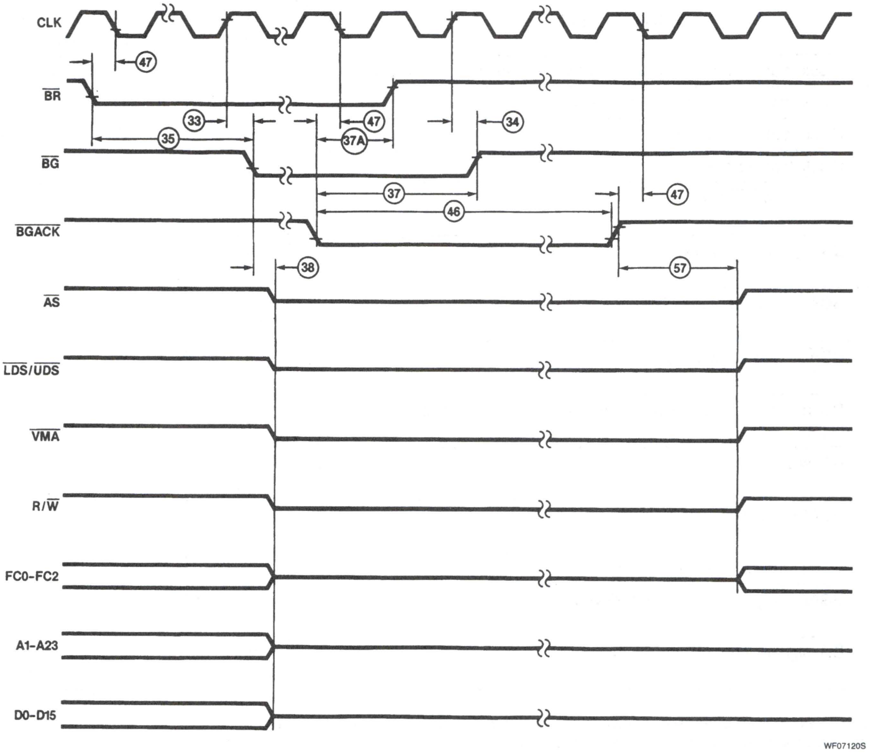

CLK
BR
BG
BGACK
AS
LDS/UDS
VMA
R/W
FC0-FC2
A1-A23
D0-D15
47
33
35
47
37A
34
37
46
47
38
57
WF07120S

Figure 57. Bus Arbitration Timing — Idle Bus Case

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

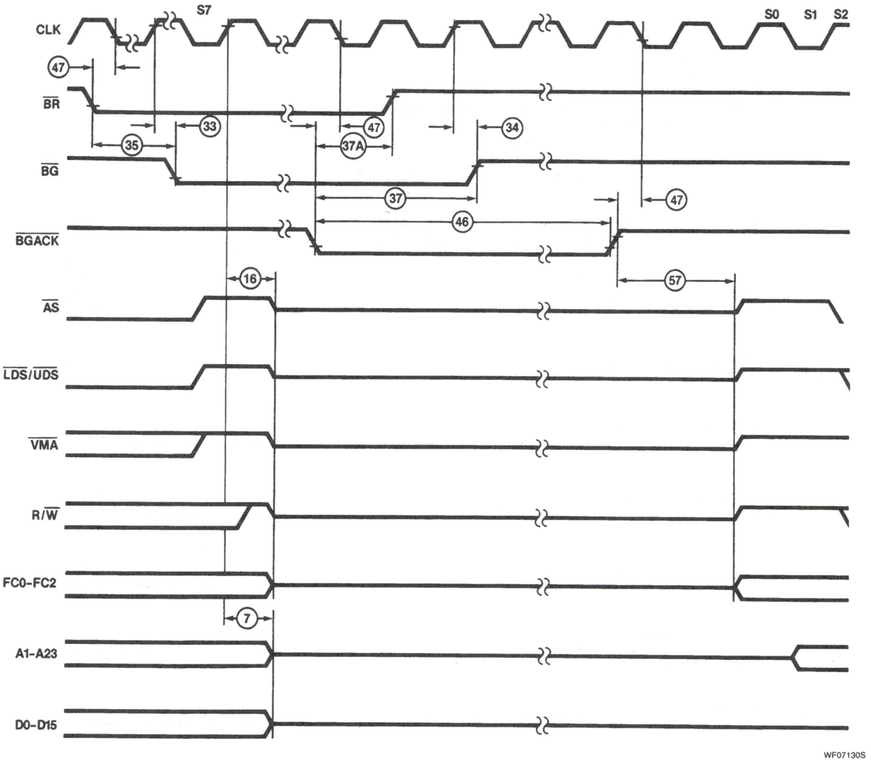

S7
S0
S1
S2
CLK
47
BR
33
35
47
37A
34
BG
37
47
46
BGACK
16
57
AS
LDS/UDS
VMA
R/W
FC0-FC2
7
A1-A23
D0-D15
WF07130S

Figure 58. Bus Arbitration Timing — Active Bus Case

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

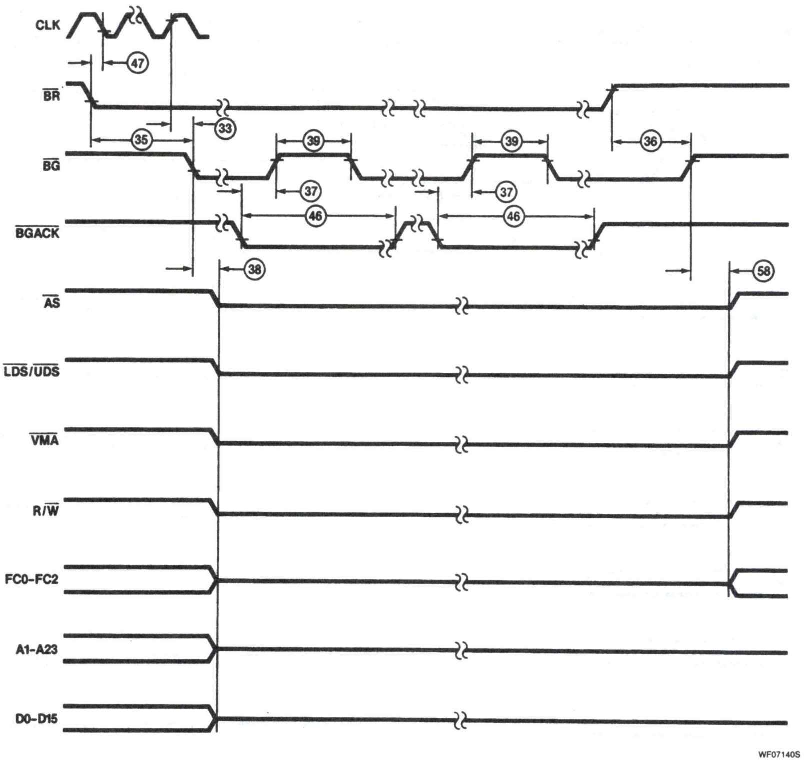

CLK
BR
BG
BGACK
AS
LDS/UDS
VMA
R/W
FC0-FC2
A1-A23
D0-D15
47
35
33
39
37
46
39
37
46
36
38
58
WF07140S

Figure 59. Bus Arbitration Timing — Multiple Bus Requests

Signetics

SCB68154
Interrupt Generator

Preliminary Specification

Microprocessor Products

DESCRIPTION

The Signetics SCB68154/8X825 Interrupt Generator provides an interface between an interrupting device and a system bus such as the VMEbus or VERSAbus®. Figure 1 shows a typical configuration of the SCB68154/8X825. The SCB68154/8X825 has three primary functions:

1. Generates bus interrupt requests.
2. Resides in the interrupt acknowledge daisy-chain.
3. Allows a status/ID byte (interrupt vector) to be supplied to the system if needed.

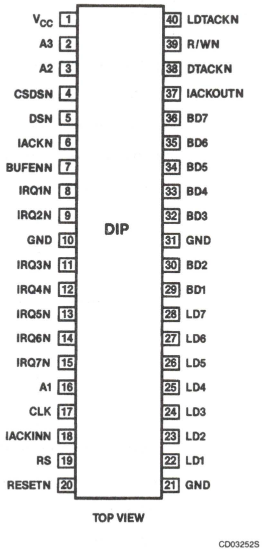
The SCB68154/8X825 has seven interrupt request levels, IRQ7N – IRQ1N, which are selected by using the interrupt request register. The local master writes to the interrupt request register to generate an interrupt request on any interrupt request level. The interrupt request register may be read to determine if an interrupt has been acknowledged. If a level with an interrupt request pending is acknowledged, the SCB68154/8X825 will allow a status/ID byte to be supplied to the system. Seven bits of the status/ID byte come from the interrupt vector register with the user externally supplying the LSB. If the SCB68154/8X825 does not have an interrupt on the level acknowledged, the SCB68154/8X825 will pass the interrupt acknowledge on via the interrupt acknowledge daisy-chain output. The user can enable all interrupt request levels and clear all interrupt request levels by setting specific bits in the interrupt vector register.

The SCB68154/8X825 was designed primarily for interface to the VMEbus. For more information regarding the protocol definitions, proper use, and application of this device, refer to the VMEbus Specification Manual.

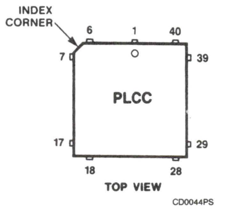
FEATURES

- **Interrupts generator for VMEbus and VERSAbus systems**
- **Generates 7 bus interrupt requests**
- **Two internal registers for system control**
- **Interrupt enable and interrupt clear bits**
- **Allows status/ID byte to be supplied during interrupt acknowledge**
- **High-speed bipolar technology**
- **Single +5V supply**

PIN CONFIGURATION



V_CC 1
A3 2
A2 3
CSDSN 4
DSN 5
IACKN 6
BUFENN 7
IRQ1N 8
IRQ2N 9
GND 10
IRQ3N 11
IRQ4N 12
IRQ5N 13
IRQ6N 14
IRQ7N 15
A1 16
CLK 17
IACKINN 18
RS 19
RESETN 20
40 LDTACKN
39 R/WN
38 DTACKN
37 IACKOUTN
36 BD7
35 BD6
34 BD5
33 BD4
32 BD3
31 GND
30 BD2
29 BD1
28 LD7
27 LD6
26 LD5
25 LD4
24 LD3
23 LD2
22 LD1
21 GND
DIP
TOP VIEW
CD03252S



INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

Pin	Signal	Pin	Signal
1	NC	23	NC
2	V_{CC}	24	GND
3	A3	25	LD1
4	A2	26	LD2
5	CSDSN	27	LD3
6	$\overline{\text{DS}}$	28	NC
7	IACKN	29	LD4
8	BUFENN	30	LD5
9	$\overline{\text{IRQ1}}$	31	LD6
10	$\overline{\text{IRQ2}}$	32	LD7
11	GND	33	BD1
12	NC	34	BD2
13	$\overline{\text{IRQ3}}$	35	GND
14	$\overline{\text{IRQ4}}$	36	BD3
15	$\overline{\text{IRQ5}}$	37	BD4
16	$\overline{\text{IRQ6}}$	38	BD5
17	$\overline{\text{IRQ7}}$	39	BD6
18	A1	40	BD7
19	CLK	41	$\overline{\text{IACKOUT}}$
20	$\overline{\text{IACKIN}}$	42	$\overline{\text{DTACK}}$
21	RS	43	$\text{R}/\overline{\text{W}}$
22	$\overline{\text{RESET}}$	44	$\overline{\text{LDTACK}}$

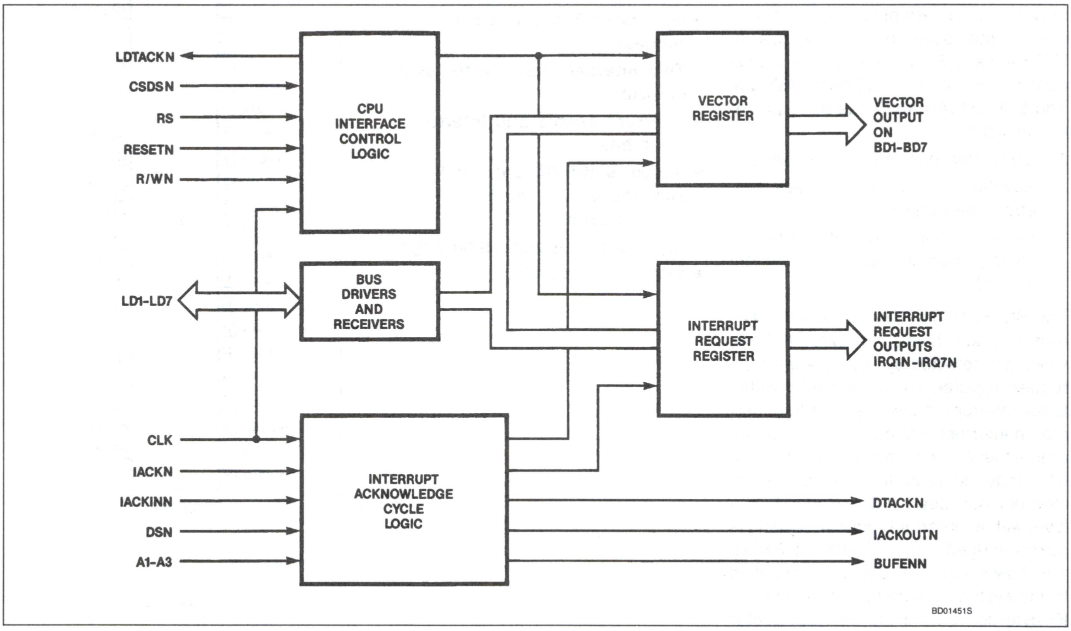
VERSAbus is a registered trademark of Motorola, Inc.

Interrupt Generator SCB68154

ORDERING INFORMATION

PACKAGES	$V_{CC} = +5V \pm 5\%$, $T_A = 0°C$ to $+70°C$
Ceramic DIP	SCB68154C2I40
Plastic DIP	SCB68154C2N40
Plastic LCC	SCB68154C2A44

BLOCK DIAGRAM


LDTACKN
CSDSN
RS
RESETN
R/WN
CPU INTERFACE CONTROL LOGIC
VECTOR REGISTER
VECTOR OUTPUT ON BD1-BD7
LD1-LD7
BUS DRIVERS AND RECEIVERS
INTERRUPT REQUEST REGISTER
INTERRUPT REQUEST OUTPUTS IRQ1N-IRQ7N
CLK
IACKN
IACKINN
DSN
A1-A3
INTERRUPT ACKNOWLEDGE CYCLE LOGIC
DTACKN
IACKOUTN
BUFENN
BD01451S

Interrupt Generator SCB68154

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V_{CC}	1	I	**Supply Voltage:** + 5V power supply
A1 – A3	16, 3, 2	I	**Address Lines:** Address inputs from system bus. The internal level being acknowledged is encoded on these inputs. A1 is LSB (least significant bit).
CSDSN	4	I	**Chip Select:** Active low chip select input for register I/O. This input must be qualified by the local master's data strobe prior to input (see Figure 1).
DSN	5	I	**Data Strobe:** Active low data strobe input from the system used to enable interrupt vector output.
IACKN	6	I	**Interrupt Acknowledge:** Active low interrupt acknowledge input from the system bus.
BUFENN	7	O	**Buffer Enable:** Active low totem pole output to enable the data buffer required to drive the outputs of the bus data pins (BD1 – BD7).
IRQ1N – IRQ7N	8, 9, 11 – 15	O	**Interrupt Request:** Active low totem pole system interrupt request output.
GND	10, 21, 31	I	**Ground**
CLK	17	I	**Clock:** Clock input (typically CPU clock).
IACKINN	18	I	**Interrupt Acknowledge In:** Active low interrupt acknowledge daisy-chain input.
RS	19	I	**Register Select:** Register select input.
RESETN	20	I	**Reset:** Active low input resets all internal registers, IACKOUTN, and IRQnN.
LD1 – LD7	22 – 28	I/O	**Local Data:** 3-State local data bus.
BD1 – BD7	29, 30, 32 – 36	O	**Bus Data:** 3-State data pins used for vector output.
IACKOUTN	37	O	**Interrupt Acknowledge Out:** Active low totem pole interrupt acknowledge daisy-chain output.
DTACKN	38	O	**Data Transfer Acknowledge:** Active low, totem pole output. This signal indicates that valid data is available on the bus during interrupt acknowledge cycle.
R/WN	39	I	**Read/Write:** Register read/write input. This signal specifies the data transfer cycle in process is to be either read or write.
LDTACKN	40	O	**Local Data Transfer Acknowledge:** Active low, open collector, data transfer acknowledge output to the local bus.

FUNCTIONAL DESCRIPTION

Typical Configuration

The SCB68154/8X825 provides a vehicle for interprocessor communications on an intelligent peripheral controller board, or a CPU board as shown in Figure 1. The local data pins (LD1 – LD7) serve as a local data bus. This allows the local master to access the Interrupt Generator's two internal registers. During an interrupt acknowledge, the SCB68154/8X825 will allow for a status/ID byte to be supplied to the system. The SCB68154/8X825 supplies seven of the eight needed status/ID bits. The user is allowed to externally supply the least significant bit (LSB), typically the system address line A1 of the status/ID byte. The IRQ1N – IRQ7N, DTACKN, and BD1 – BD7 outputs require external buffers to provide adequate drive to the system bus. BUFENN provides the output enable control for the data buffer that is required for BD1 – BD7.

Register Selection

The SCB68154/8X825 has two internal registers which can be programmed for system control. They are the interrupt vector register and the interrupt request register. Figure 2 shows the programming model. Both registers can be read from as well as written to. The interrupt vector register (register R0) is selected by the register select (RS) input equal to 0. Setting bit 1 of register R0 enables all interrupt levels for the SCB68154/8X825. Writing a 1 to bit 2 of register R0 resets all interrupt levels in the interrupt request register as well as the IRQnN outputs. Subsequent interrupt requests will be honored. Bit 2 of R0 will always be read as 0. The high order bits, bits 7 – 3, of register R0 are the high order bits of the status/ID byte. The seven bit output of the status/ID byte are formed by concatenating the high order bits (bits 7 – 3) of register R0 with system bus address lines A3 and A2. Bus address lines A3 and A2 are output on BD2 and BD1 respectively.

The interrupt request register (R1) is selected by RS input equal to 1. Setting bit "n" in R1 will generate an interrupt on interrupt request level IRQnN. Any number (up to the maximum of seven) interrupt requests can be generated in a single access of R1.

The state of only those levels, which a 1 has been written to, is affected. Writing a 0 to any level does not change the current state of that level. For example, if IRQ1N is currently asserted, writing a 0 to bit 1 of R1 does not de-assert IRQ1N, nor clear bit 1 in R1.

Note that interrupt requests on the same level are not stackable. To generate another interrupt request on a level currently asserted, the user must wait until that level has been acknowledged, before generating another interrupt request on that level.

Since interrupts are acknowledged independently of the local CPU, the interrupt vector register should not be modified while interrupts are pending. Any attempt to modify the interrupt vector register, while interrupts are pending, could cause the status/ID byte to change while the interrupted master is reading it. This could cause the interrupted master to acquire an indeterminate vector. Therefore, the interrupt request register should be examined to make certain there are no interrupts pending before attempting to modify the interrupt vector register.

All data transfers between the local CPU and the SCB68154/8X825 are done using local data lines (LD1 – LD7), register select (RS), read/write (R/WN) and a chip select input (CSDSN). The SCB68154/8X825 supplies a local data transfer acknowledge (LDTACKN) to complete the transfer of data between the local CPU and itself.

Interrupt/Interrupt Acknowledge

The SCB68154/8X825 generates the maximum defined seven bus interrupts, on the IRQ1N – IRQ7N outputs, in a VMEbus or VERSAbus system. An interrupted master will acknowledge only a single level of the seven interrupt levels. To allow for multiple interrupters on the level acknowledged, VMEbus and VERSAbus systems use an interrupt acknowledge daisy-chain. The SCB68154/8X825 resides in this interrupt acknowledge daisy-chain.

When the system interrupt acknowledge (IACKN) is asserted, the interrupt acknowledge daisy-chain starts at the first slot in the system bus. The level being acknowledged is specified by the interrupted master on address lines A1 – A3. The system bus interface on the SCB68154/8X825 is initiated only if IACKN, interrupt acknowledge daisy-chain in (IACKINN), and data strobe 0 (DS0N) are all received asserted. If the system bus interface is initiated and the SCB68154/8X825 has an interrupt request on the level specified, it will not pass the daisy-chain signal on. It will, instead, clear the interrupt request level acknowledged, IRQnN, as well as the appropriate bit in the interrupt request register. It will also assert buffer enable (BUFENN), place seven bits of the status/ID byte on the bus data outputs (BD1 – BD7) and assert data transfer acknowledge (DTACKN).

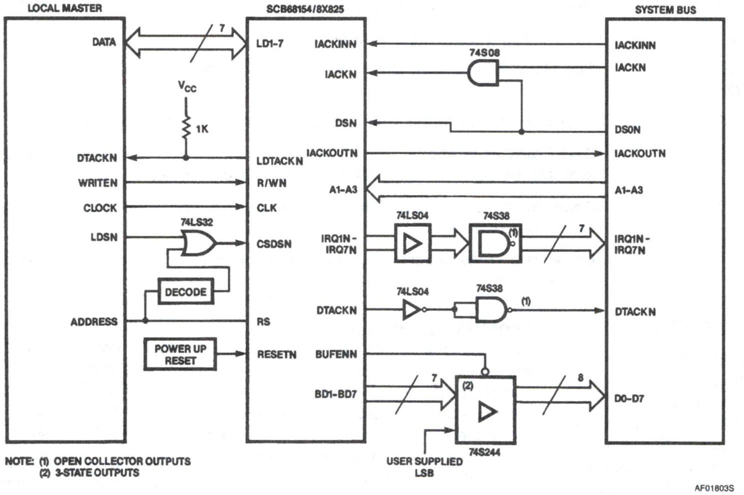
If the system bus interface is initiated, but the SCB68154/8X825 has no interrupt request on the level being acknowledged, it will pass the daisy-chain input (IACKINN) on via the interrupt acknowledge out (IACKOUTN).

Arbitration

The system bus interface, as well as the local master interface, are independent processes. Either can be initiated at any time, without respect to the other. The SCB68154/8X825 will arbitrate between the processes, allowing proper system operation without any degradation in performance.

Reset

When RESETN is asserted, the SCB68154/8X825 drives the outputs IRQnN and IACKOUTN high. It also resets the interrupt request and the interrupt vector registers.


LOCAL MASTER
DATA
7
VCC
1K
DTACKN
WRITEN
CLOCK
LDSN
74LS32
DECODE
ADDRESS
POWER UP RESET
SCB68154/8X825
LD1-7
IACKINN
IACKN
DSN
LDTACKN
IACKOUTN
R/WN
A1-A3
CLK
CSDSN
IRQ1N-IRQ7N
RS
DTACKN
RESETN
BUFENN
BD1-BD7
74S08
74LS04
74S38
(1)
7
74LS04
74S38
(1)
7
(2)
8
74S244
USER SUPPLIED LSB
SYSTEM BUS
IACKINN
IACKN
DSON
IACKOUTN
A1-A3
IRQ1N-IRQ7N
DTACKN
D0-D7
AF01803S

NOTE: (1) OPEN COLLECTOR OUTPUTS
(2) 3-STATE OUTPUTS

Figure 1. Typical Configuration

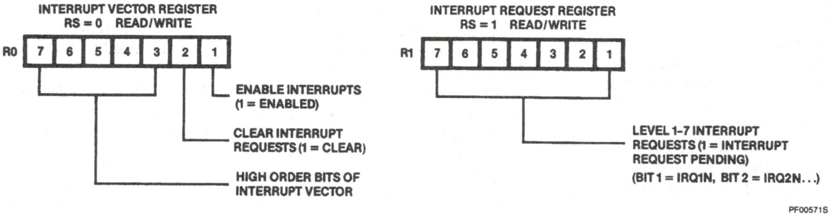

INTERRUPT VECTOR REGISTER
RS = 0 READ/WRITE
R0
7 6 5 4 3 2 1
ENABLE INTERRUPTS (1 = ENABLED)
CLEAR INTERRUPT REQUESTS (1 = CLEAR)
HIGH ORDER BITS OF INTERRUPT VECTOR
INTERRUPT REQUEST REGISTER
RS = 1 READ/WRITE
R1
7 6 5 4 3 2 1
LEVEL 1-7 INTERRUPT REQUESTS (1 = INTERRUPT REQUEST PENDING)
(BIT 1 = IRQ1N, BIT 2 = IRQ2N...)
PF00571S

Figure 2. Programming Model

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
V_{CC}	Supply voltage[3] range	−0.5 to +7.0	V
V_{IN}	Input voltage[3] range	−0.5 to +5.5	V
V_{OUT}	Voltage applied to output in off-state[3]	−0.5 to +5.5	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$[4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{CC}	Supply voltage		4.75	5.25	V
V_{IL}	Input low voltage			0.8	V
V_{IH}	Input high voltage		2		V
I_{IL}	Input low current	$V_{CC} = 5.25V$, $V_{IL} = 0.4V$		−410	μA
I_{IH}	Input high current	$V_{CC} = 5.25V$, $V_{IH} = 2.7V$		20	μA
I_{OS}	Short circuit output current	$V_{CC} = 5.25V$, $V_{OUT} = 0V$ Other outputs not grounded	−15	−100	mA
I_{OZL}	High-Z low output current BD1 – BD7	$V_{CC} = 5.25V$, $V_{OL} = 0.5V$		−20	μA
I_{OZH}	High-Z high output current BD1 – BD7	$V_{CC} = 5.25V$, $V_{OH} = 2.5V$		20	μA
V_{OL}	Output low voltage			0.5	V
	LDTACKN	$V_{CC} = 4.75V$, $I_{OL} = 20mA$			
	All other outputs	$V_{CC} = 4.75V$, $I_{OL} = 8mA$			
V_{OH}	Output high voltage	$V_{CC} = 4.75V$, $I_{OH} = -400\mu A$	2.5		V
I_{CEX}	Open collector leakage current LDTACKN	$V_{CC} = 4.75V$, $V_{OUT} = 4.75V$		100	μA
I_I	Input leakage current	$V_{CC} = 5.25V$, $V_{IN} = 5.25V$		100	μA
V_{IC}	Input clamp voltage	$V_{CC} = 4.75V$, $I_{IN} = -10mA$	−1.2		V
I_{CC}	Supply current	$V_{CC} = 5.25V$, $V_{IN} = 0V$		130	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient for ceramic package (116°C/W for plastic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all inputs swing between 0.4V and 2.4V with a transition time of 3ns maximum and output voltages are checked at 0.8V and 2V.
6. If the falling edge of IACKINN occurs last, then t_{ADIK} is valid. If the falling edge of DSN occurs last, then t_{ADDS} is valid. If the falling edges of IACKINN and DSN occur simultaneously then either t_{ADDS} or t_{ADIK} is valid.
7. If the falling edge of IACKINN occurs last, then t_{IKBF} is valid. If the falling edge of DSN occurs last, then t_{DSBF} is valid. If the falling edges of IACKINN and DSN occur simultaneously then either t_{IKBF} or t_{DSBF} is valid.
8. If the falling edge of IACKINN occurs last, then t_{IRA1} is valid. If DSN occurs last, then t_{IRA2} is valid. If the falling edges of both IACKINN and DSN occur simultaneously then either t_{IRA1} or t_{IRA2} is valid.
9. True only if no request pending on levels being acknowledged. If the falling edge of IACKINN occurs last, then t_{DYO1} is valid. If the falling edge of DSN occurs last then t_{DYO2} is valid. If the falling edges of both IACKINN and DSN occur simultaneously then either t_{DYO1} or t_{DYO2} is valid.
10. If the rising edge of IACKN occurs first, then t_{IDT} is valid. If the rising edge of DSN occurs first then t_{DSDT} is valid. If the rising edges of both IACKN and DSN occur simultaneously, then either t_{IDT} or t_{DSDT} is valid.
11. If the rising edge of IACKN occurs first then t_{ITST} is valid. If the rising edge of DSN occurs first then t_{DTST} is valid. If the rising edges of both IACKN and DSN occur simultaneously then either t_{ITST} or t_{DTST} is valid.
12. If the rising edge of IACKN occurs first, then t_{IBF} is valid. If the rising edge of DSN occurs first then t_{DBF} is valid. If the rising edges of both IACKN and DSN occur simultaneously, then either t_{IBF} or t_{DBF} is valid.
13. If the rising edge of IACKN occurs first then t_{IDTK} is valid. If the rising edge of DSN occurs first then t_{DDTK} is valid. If the rising edges of both IACKN and DSN occur simultaneously, then either t_{IDTK} or t_{DDTK} is valid.
14. True only if no request pending on level being acknowledged. If the rising edge of IACKN occurs first then t_{IOUT} is valid. If the rising edge of DSN occurs first then t_{DOUT} is valid. If the rising edge of both IACKN and DSN occur simultaneously then either t_{IOUT} or t_{DOUT} is valid.
15. t_{TST} is always greater than or equal to t_{DTH}.

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$[4, 5]

SYMBOL	PARAMETER	TENTATIVE LIMITS		UNIT
		Min	Max	
Register read (see Figure 3)				
t_{RSS}	RS valid to CSDSN low setup time	0		ns
t_{RWS}	R/WN to CSDSN low setup time	0		ns
t_{DTV}	LDTACKN low to LD1 – 7 valid	5.4	19.4	ns
t_{DTCS}	LDTACKN low to CSDSN high	0		ns
t_{RSH}	CSDSN high to RS valid hold time	0		ns
t_{RWH}	CSDSN high to R/WN high hold time	0		ns
t_{DTH}	CSDSN high to LD1 – 7 valid hold time	11.5	33.5	ns
t_{TST}[15]	CSDSN high to LD1 – 7 3-State	11.5	34.5	ns
t_{CSDT}	CSDSN high to LDTACKN high	9.2	27.2	ns
t_{CSH}	CSDSN high time	20		ns
t_{ACCR}	CSDSN low to LDTACKN low read access time	$t_{CKPD} \div 2 + 12.5$	$3t_{CKPD} \div 2 + 40.3$	ns
Register write (see Figure 4)				
t_{RSS}	RS valid to CSDSN low setup time	0		ns
t_{RWS2}	R/WN low to CSDSN low setup time	0		ns
t_{DS}	LD1 – LD7 valid to CSDSN low setup time	0		ns
t_{RSH}	CSDSN high to RS valid hold time	0		ns
t_{RWH2}	CSDSN high to R/WN low hold time	0		ns
t_{DH}	CSDSN high to LD1 – LD7 valid	0		ns
t_{CSDT}	CSDSN high to LDTACKN high	9.2	27.2	ns
t_{DTCS}	LDTACKN low to CSDSN high	0		ns
t_{IRQ}	LDTACKN low to IRQnN low	1.2	8.2	ns
t_{ACCW}	CSDSN low to LDTACKN low write access time	$t_{CKPD} \div 2 + 13.3$	$3t_{CKPD} \div 2 + 40.3$	ns
t_{CSH}	CSDSN high time	20		ns
Interrupt acknowledge (see Figure 5)				
t_{IKDS}	IACKN low to DSN low	0		ns
t_{ADDS}[6]	A1 – A3 valid to DSN low setup	0		ns
t_{ADIK}[6]	A1 – A3 valid to IACKINN low setup	0		ns
t_{IKBF}[7]	IACKINN low to BUFENN low	$t_{CKPD} + 18.3$	$3t_{CKPD} \div 2 + 29.2$	ns
t_{DSBF}[7]	DSN low to BUFENN low	$t_{CKPD} + 18.3$	$3t_{CKPD} \div 2 + 29.2$	ns
t_{IRA1}[8]	IACKINN low to IRQnN high	$t_{CKPD} + 12.5$	$3t_{CKPD} \div 2 + 29.2$	ns
t_{IRA2}[8]	DSN low to IRQnN high	$t_{CKPD} + 29.2$	$3t_{CKPD} \div 2 + 29.2$	ns
t_{DYO1}[9]	IACKINN low to IACKOUTN	$t_{CKPD} + 11.2$	$3t_{CKPD} \div 2 + 29.2$	ns
t_{DYO2}[9]	DSN low to IACKOUTN low	$t_{CKPD} + 11.2$	$3t_{CKPD} \div 2 + 29.2$	ns

AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	TENTATIVE LIMITS		UNIT
		Min	Max	
t_{BFBD}	BUFENN low to BD1 – BD7 valid	3.2	38.2	ns
t_{BFDT}	BUFENN low to DTACKN low	38.2	50	ns
t_{ADRH}	DTACKN low to A1 – A3 valid hold time	0		ns
t_{DTDS}	DTACKN low to DSN high	0		ns
t_{DTIK}	DTACKN low to IACKN high	0		ns
t_{IDT}[10]	IACKN high to BD1 – BD7 valid hold time	3	8.4	ns
t_{DSDT}[10]	DSN high to BD1 – BD7 valid hold time	3	8.4	ns
t_{ITST}[11]	IACKN high to BD1 – BD7 3-State	3	9.1	ns
t_{DTST}[11]	DSN high to BD1 – BD7 3-State	3	9.1	ns
t_{DBF}[12]	DSN high to BUFENN high	12.2	32.2	ns
t_{IBF}[12]	IACKN high to BUFENN high	12.2	32.2	ns
t_{DDTK}[13]	DSN high to DTACKN high	12.2	32.2	ns
t_{IDTK}[13]	IACKN high to DTACKN high	12.2	32.2	ns
t_{IOUT}[14]	IACKN high to IACKOUTN high	6.2	17.2	ns
t_{DOUT}[14]	DSN high to IACKOUTN high	6.2	17.2	ns
t_{IAKH}	IACKN high time	20		ns
t_{IINH}	IACKN high to IACKINN high	0		ns
t_{IKIN}	IACKN low to IACKINN low	0		ns
Reset timing (see Figure 6)				
t_{RST}	RESETN low time	51		ns
Clock timing (see Figure 7)				
t_{CKH}	Clock high	45		ns
t_{CKPD}	Clock period	90		ns

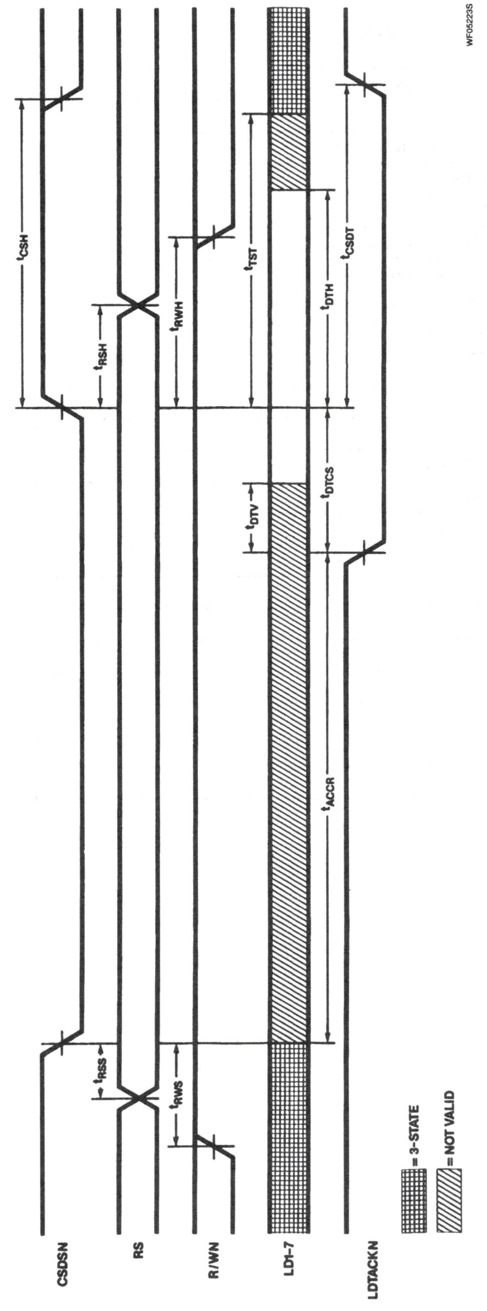

CSDSN
RS
R/WN
LD1–7
LDTACKN
tCSH
tRSH
tRWH
tTST
tDTH
tCSDT
tDTCS
tDTV
tACCR
tRSS
tRWS
▦ = 3-STATE
▨ = NOT VALID
WF05223S

Figure 3. Register Read

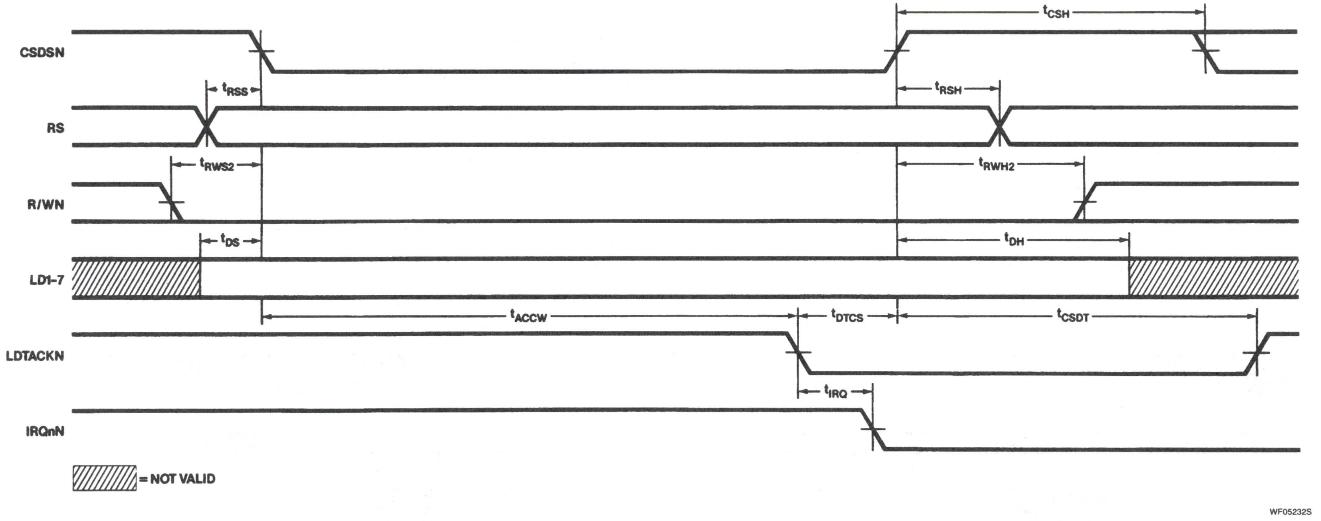

CSDSN
RS
R/WN
LD1-7
LDTACKN
IRQnN
t_{CSH}
t_{RSS}
t_{RSH}
t_{RWS2}
t_{RWH2}
t_{DS}
t_{DH}
t_{ACCW}
t_{DTCS}
t_{CSDT}
t_{IRQ}
= NOT VALID
WF05232S

Figure 4. Register Write

Interrupt Generator

SCB68154

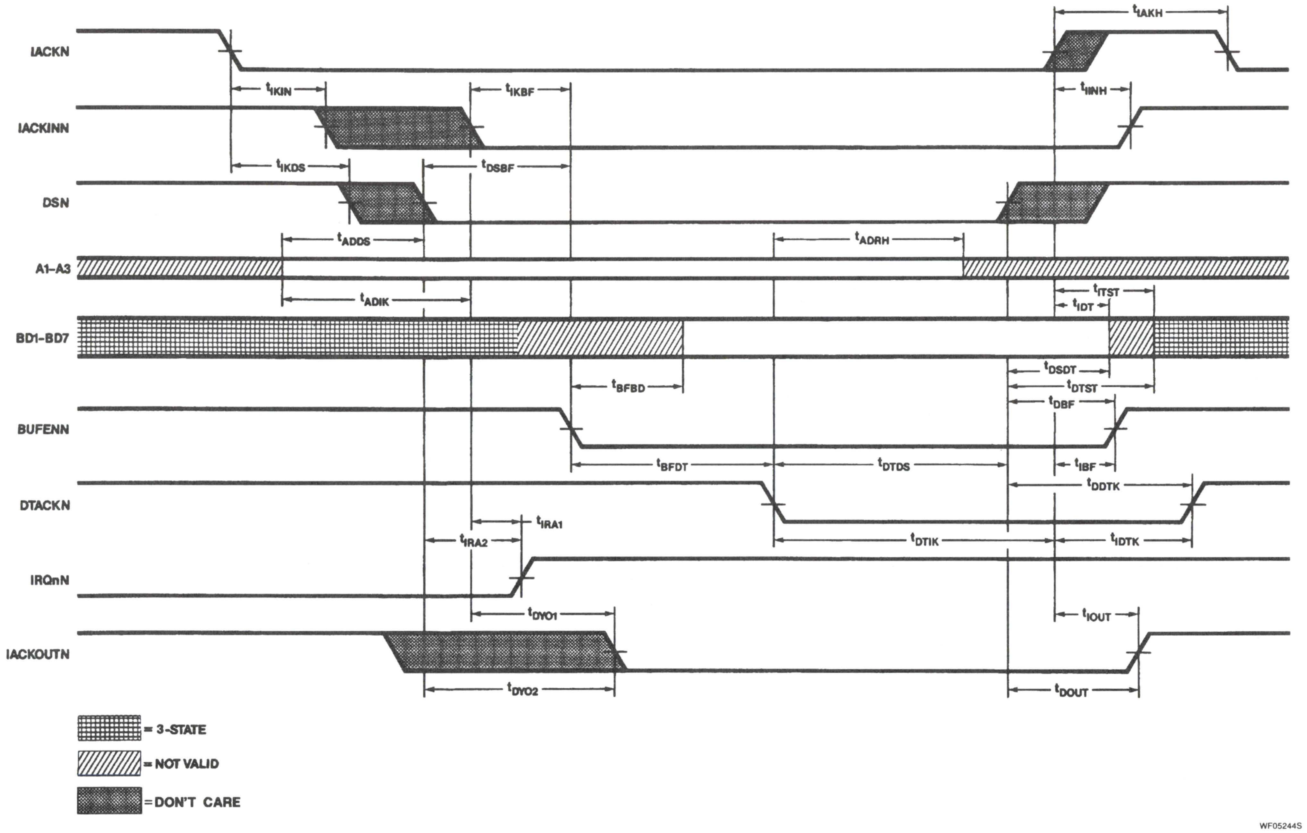

IACKN
IACKINN
DSN
A1–A3
BD1–BD7
BUFENN
DTACKN
IRQnN
IACKOUTN
t_{AKH}
t_{IKIN}
t_{IKBF}
t_{IINH}
t_{IKDS}
t_{DSBF}
t_{ADDS}
t_{ADRH}
t_{ADIK}
t_{ITST}
t_{IDT}
t_{DSDT}
t_{DTST}
t_{BFBD}
t_{DBF}
t_{BFDT}
t_{DTDS}
t_{IBF}
t_{DDTK}
t_{IRA1}
t_{IRA2}
t_{DTIK}
t_{IDTK}
t_{DYO1}
t_{IOUT}
t_{DYO2}
t_{DOUT}
= 3-STATE
= NOT VALID
= DON'T CARE
WF05244S

Figure 5. Interrupt Acknowledge

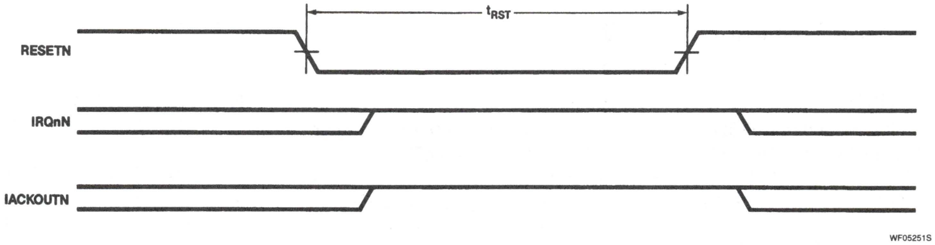

t_{RST}
RESETN
IRQnN
IACKOUTN
WF05251S

Figure 6. Reset Timing

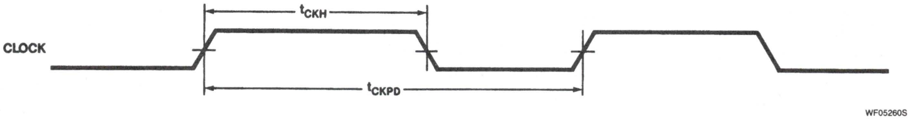

tCKH
CLOCK
tCKPD
WF05260S

Figure 7. Clock Timing

Signetics

SCB68155
Interrupt Handler

Preliminary Specification

Microprocessor Products

DESCRIPTION

The Signetics SCB68155/8X824 is an asynchronous interrupt handler for VMEbus and VERSAbus® systems. Up to 14 interrupts are prioritized by the SCB68155/8X824 to one of seven levels and are output on the interrupt priority level lines (IPL0N – IPL2N). The SCB68155/8X824 prioritizes the interrupts in the following manner: local bus requests over system bus requests with the non-maskable interrupt (NMIN) considered the highest priority local interrupt (NMIN over IRQ7N, then LRQ6N – LRQ1N over IRQ6N – IRQ1N).

The local interrupt requests can be programmed to be either active high or low, and either edge or level sensitive. The system bus interrupt requests are always active low and level sensitive. The non-maskable interrupt is always negative edge-triggered.

During a local interrupt acknowledge sequence, two modes of response are available: vectored mode or device-supplies-the vector mode.

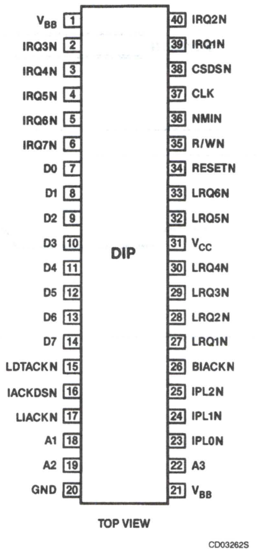
For system bus responses, the SCB68155/8X824 works with a bus requester (for example, the Signetics SCB68172 VMEbus Controller), to acquire a status/ID byte (interrupt vector) from the system.

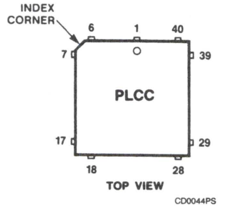
The SCB68155/8X824 was designed primarily for interface to the VMEbus. For more information regarding the protocol definitions, proper use, and application of this device, refer to the VMEbus Specification Manual.

FEATURES

- **Asynchronous interrupt handler for VMEbus and VERSAbus systems**
- **Receives and prioritizes non-maskable, six local and seven system bus interrupts**
- **Interrupts may be polled in lieu of real-time operation**
- **Programmable local interrupt response**
- **Works with the SCB68172 to acquire status/ID byte (vector) during bus interrupt acknowledge**
- **Complete device status, including last interrupt acknowledged**
- **High-speed bipolar technology**

PIN CONFIGURATION


V$_{BB}$ 1
IRQ3N 2
IRQ4N 3
IRQ5N 4
IRQ6N 5
IRQ7N 6
D0 7
D1 8
D2 9
D3 10
D4 11
D5 12
D6 13
D7 14
LDTACKN 15
IACKDSN 16
LIACKN 17
A1 18
A2 19
GND 20
40 IRQ2N
39 IRQ1N
38 CSDSN
37 CLK
36 NMIN
35 R/WN
34 RESETN
33 LRQ6N
32 LRQ5N
31 V$_{CC}$
30 LRQ4N
29 LRQ3N
28 LRQ2N
27 LRQ1N
26 BIACKN
25 IPL2N
24 IPL1N
23 IPL0N
22 A3
21 V$_{BB}$
DIP
TOP VIEW
CD03262S


INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

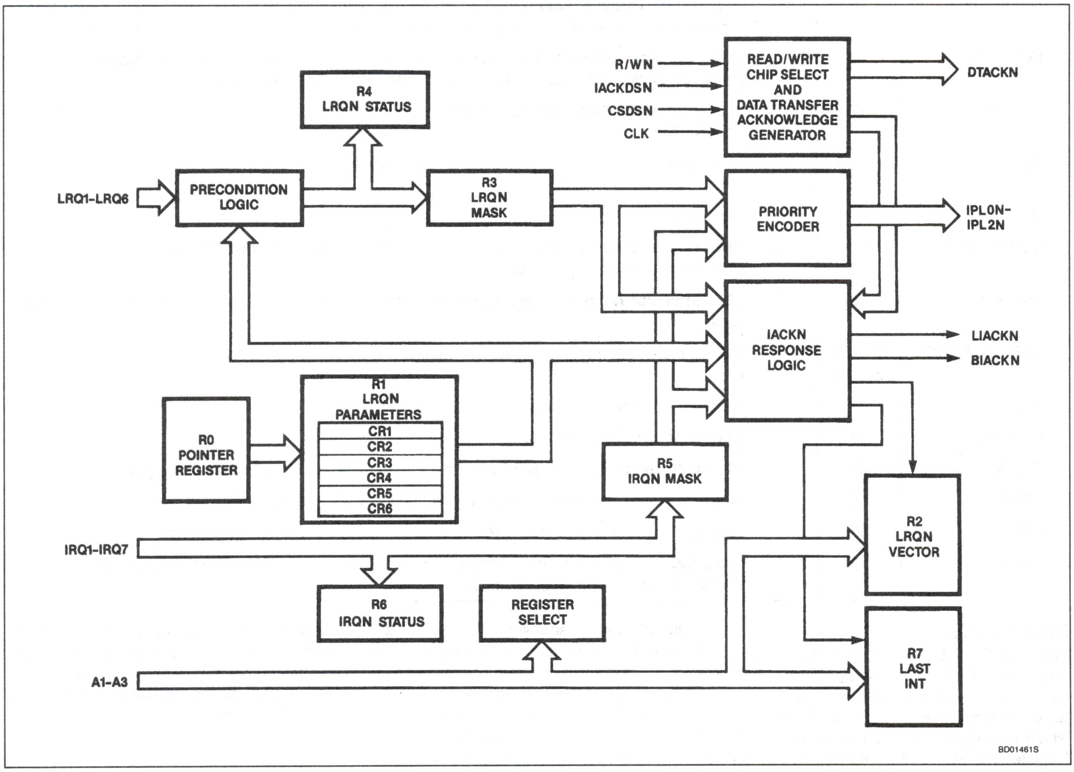
Pin	Signal	Pin	Signal
1	VBB	23	GND
2	IRQ3N	24	VBB
3	IRQ4N	25	A3
4	IRQ5N	26	$\overline{\text{IPL0}}$
5	IRQ6N	27	$\overline{\text{IPL1}}$
6	IRQ7N	28	$\overline{\text{IPL2}}$
7	NC	29	$\overline{\text{BIACK}}$
8	D0	30	NC
9	D1	31	$\overline{\text{LRQ1}}$
10	D2	32	$\overline{\text{LRQ2}}$
11	D3	33	$\overline{\text{LRQ3}}$
12	D4	34	$\overline{\text{LRQ4}}$
13	D5	35	V_{CC}
14	D6	36	$\overline{\text{LRQ5}}$
15	D7	37	$\overline{\text{LRQ6}}$
16	NC	38	$\overline{\text{RESET}}$
17	NC	39	R/$\overline{\text{W}}$
18	$\overline{\text{LDTACK}}$	40	$\overline{\text{NMI}}$
19	$\overline{\text{IACKDS}}$	41	CLK
20	$\overline{\text{LIACK}}$	42	$\overline{\text{CSDS}}$
21	A1	43	$\overline{\text{IRQ1}}$
22	A2	44	$\overline{\text{IRQ2}}$

VERSAbus is a registered trademark of Motorola, Inc.

ORDERING INFORMATION

PACKAGES	$V_{CC} = +5V \pm 5\%$, $T_A = 0°C$ to $+70°C$
Ceramic DIP	SCB68155CAI40
Plastic DIP	SCB68155CAN40
Plastic LCC	SCB68155CA44

BLOCK DIAGRAM


R/WN
IACKDSN
CSDSN
CLK
READ/WRITE CHIP SELECT AND DATA TRANSFER ACKNOWLEDGE GENERATOR
DTACKN
R4 LRQN STATUS
LRQ1-LRQ6
PRECONDITION LOGIC
R3 LRQN MASK
PRIORITY ENCODER
IPL0N-IPL2N
IACKN RESPONSE LOGIC
LIACKN
BIACKN
R0 POINTER REGISTER
R1 LRQN PARAMETERS
CR1
CR2
CR3
CR4
CR5
CR6
R5 IRQN MASK
R2 LRQN VECTOR
IRQ1-IRQ7
R6 IRQN STATUS
REGISTER SELECT
R7 LAST INT
A1-A3
BD01461S

PIN DESCRIPTION

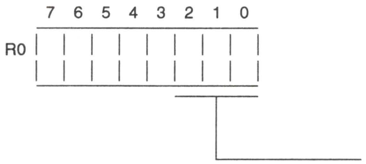
MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V_{BB}	1, 21	I	**Supply Voltage:** Supply voltage for internal gates.
IRQ1N – IRQ7N	39, 40, 2 – 6	I	**Bus Interrupt Request:** Active low inputs for bus generated interrupts.
D0 – D7	7 – 14	I/O	**Bus Data:** 3-State local data bus.
LDTACKN	15	O	**Local Data Transfer Acknowledge:** Active low, open collector output. Indicates that valid data is available on the local data bus during interrupt acknowledge cycle or data transfer cycle.
IACKDSN	16	I	**Interrupt Acknowledge:** Active low interrupt acknowledge input from the local master. This signal must be qualified by the local master's data strobe prior to input.
LIACKN	17	O	**Local Interrupt Acknowledge:** Active low interrupt acknowledge totem pole output to the local interrupting devices.
A1 – A3	18, 19, 22	I	**Address Lines:** Address inputs from local master.
GND	20	I	**Ground**
IPL0N – IPL2N	23 – 25	O	**Interrupt Priority Level:** Active low totem-pole outputs to the local master. The priority level of the interrupt request is encoded on these outputs.
BIACKN	26	O	**Bus Interrupt Acknowledge:** Active low interrupt acknowledge totem-pole output to the system bus.
LRQ1N – LRQ6N	27 – 30, 32, 33	I	**Local Interrupt Request:** User can define the active state of these inputs.
V_{CC}	31	I	**Supply Voltage:** +5V power supply.
RESETN	34	I	**Reset:** Active low input reset.
R/WN	35	I	**Read/Write:** This signal specifies the data transfer cycle to be either read or write.
NMIN	36	I	**Non-Maskable Interrupt:** Active low highest priority interrupt.
CLK	37	I	**Clock:** Clock input (typically CPU clock).
CSDSN	38	I	**Chip Select:** Active low chip select input for register I/O. This input must be qualified by the local master's data strobe prior to input.

REGISTERS

The SCB68155/8X824 contains eight internal registers (R0 – R7) accessible to the local master. There are also six sub-registers contained in register R1. Register R0 specifies which sub-register is to be accessed in R1. Register R2 stores the interrupt vector for vectored mode responses. Register R3 and R5 are the interrupt mask registers for the local and system bus interrupts respectively. Registers R4 and R6 are the status registers for local and bus interrupts respectively, allowing all interrupts to be polled. Register R7 can be read by the local master to determine the last interrupt acknowledged.

All data transfers between the SCB68155/8X824 and the local master are done using the local data bus (D0 – D7), address bus (A1 – A3), a chip select (CSDSN) and a read/write (R/WN) input.

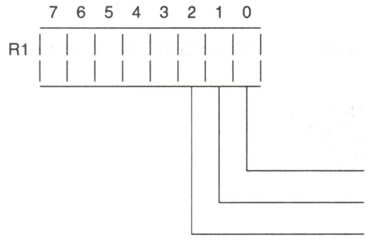
Register R0 — A3A2A1 = 000



Pointer register (write only).
Bit 0 – 2 of R0 specify which control sub-register CR1 – CR6 during an access of R1.
During register I/O, bits 7 – 3 will read as 0.

B2 – B1 – B0
000 – none
001 – CR1
010 – CR2
011 – CR3
100 – CR4
101 – CR5
110 – CR6
111 none

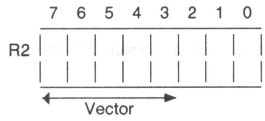
Register R1 — A3A2A1 = 001



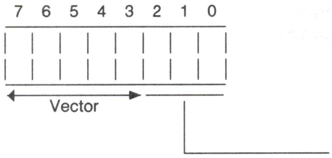
Control registers CR1 – CR6 (read or write).
These six registers program the function of the local interrupt requests (LRQ1N – LRQ6N). (CR1 programs LRQ1N, CR2 programs LRQ2N, etc). During register I/O, bits 7 – 3 will be read as 0.

LRQnN active state (high/low)	(1 = active high)
LRQnN edge/level sensitive	(1 = edge sensitive)
LRQnN vector enable	(1 = enabled)

Register R2 — A3A2A1 = 010



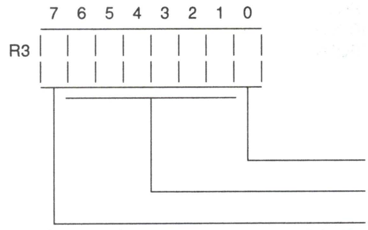
LRQ vector (read or write).
Bits 7 – 3 of this register are the top five bits of the local interrupt vector. During register I/O, bits 2 – 0 will be read as zeros.



LRQ vector output during local interrupt acknowledge (If vector enable = 1).

001 – LRQ1N
010 – LRQ2N
011 – LRQ3N
100 – LRQ4N
101 – LRQ5N
110 – LRQ6N
111 – NMIN

Register R3 — A3A2A1 = 011

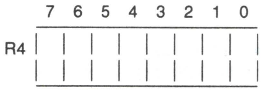


LRQ mask (read or write).
This register allows the user to mask local interrupts. It also enables vectored response for NMIN.

NMIN Vector enable	(1 = enabled)
LRQN 1 – 6 mask	(1 = interrupt enabled)
NMIN mask	(1 = NMIN enabled)

Bit 1 = LRQ1N
Bit 2 = LRQ2N
Bit 3 = LRQ3N
Bit 4 = LRQ4N
Bit 5 = LRQ5N
Bit 6 = LRQ6N
Bit 7 = NMIN

Register R4 — A3A2A1 = 100

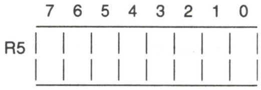


LRQ status (read only).
Local interrupts can be polled through this register. During register I/O, bit 0 will be read as a 0.

Bit 1 = LRQ1N
Bit 2 = LRQ2N
Bit 3 = LRQ3N
Bit 4 = LRQ4N
Bit 5 = LRQ5N
Bit 6 = LRQ6N
Bit 7 = NMIN

LRQN status (1 = interrupt pending)
NMIN status (1 = interrupt pending)

Register R5 — A3A2A1 = 101

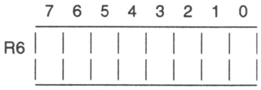


IRQ mask (read or write).
This register allows the user to mask system bus interrupts. During register I/O, bit 0 will be read as a 0.

Bit 1 = IRQ1N
Bit 2 = IRQ2N
Bit 3 = IRQ3N
Bit 4 = IRQ4N
Bit 5 = IRQ5N
Bit 6 = IRQ6N
Bit 7 = IRQ7N

(1 = interrupt enabled)

Register R6 — A3A2A1 = 110

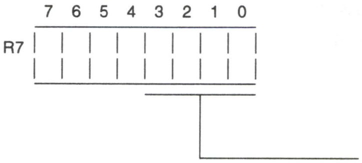


IRQ status (read only).
System bus interrupts can be polled through this register. During register I/O, bit 0 will be read as 0.

Bit 1 = IRQ1N
Bit 2 = IRQ2N
Bit 3 = IRQ3N
Bit 4 = IRQ4N
Bit 5 = IRQ5N
Bit 6 = IRQ6N
Bit 7 = IRQ7N

IRQN status (1 = interrupt pending)

Register R7 — A3A2A1 = 111



Last interrupt acknowledged (read only).
This register can be read by the local CPU to determine the last interrupt acknowledged. During register I/O, bits 7 – 4 will be read as 0.

0000 – none
0001 – IRQ1N
0010 – IRQ2N
0011 – IRQ3N
0100 – IRQ4N
0101 – IRQ5N
0110 – IRQ6N
0111 – IRQ7N
1000 – none
1001 – LRQ1N
1010 – LRQ2N
1011 – LRQ3N
1100 – LRQ4N
1101 – LRQ5N
1110 – LRQ6N
1111 – NMIN

Interrupt Handler SCB68155

FUNCTIONAL OPERATION

Typical Configuration

The SCB68155/8X824 can handle interrupts from 14 sources: seven bus interrupt requests generated on the IRQ1N – IRQ7N inputs, six local interrupt sources generated on the LRQ1N – LRQ6N inputs, and one non-maskable interrupt which may originate locally or from the system (such as the system's AC fail signal). All interrupts are encoded to one of seven levels and output on the IPL0N – IPL2N lines. Table 1 shows how the SCB68155/8X824 encodes the interrupts.

BIACKN is the bus interrupt acknowledge signal which is asserted during a bus interrupt acknowledge sequence. BIACKN can be used to get the associated bus requester (for example, the Signetics SCB68172), to acquire an interrupt vector from the system bus. Figure 1 shows a typical SCB68155/8X824-SCB68172 system configuration.

LIACKN is the local interrupt acknowledge signal which is asserted during a local interrupt acknowledge sequence. Figure 2 shows a typical configuration for the SCB68155/8X824.

Non-Maskable Interrupt (NMI)

The highest priority interrupt request is the non-maskable interrupt (NMIN). It is an active low, negative edge-triggered interrupt. NMIN is considered by the SCB68155/8X824 to be the highest priority local interrupt, however, the user is not restricted to having it represent a local device. When the local master responds to an NMIN, bit 7 in the LRQ status register R4 is cleared to 0.

Both vectored and device-supplies-the-vector modes are available with NMIN. However, it is recommended that the SCB68155/8X824's response to an NMIN be a vectored mode interrupt acknowledge.

Local Interrupts

The SCB68155/8X824 can handle interrupts generated by local devices through its six local interrupt request lines (LRQ1N – LRQ6N). The local interrupt requests are prioritized with LRQ6N being the highest priority, and LRQ1N the lowest priority.

The response of the SCB68155/8X824 to an acknowledge of a local interrupt can be selected by means of the SCB68155/8X824's R1 register. Pointer register R0 points to one of the six control sub-registers when accessing register R1. The six control registers (CR1 – CR6) in register R1 define the functions of the six local interrupt requests (LRQ1N – LRQ6N).

Control Register 'n' Bit 0

Selects local interrupt requests 'n' (LRQnN), to be either low or high. Bit 0 = 1 defines active state to be high.

Control Register 'n' Bit 1

Selects local interrupt request 'n', to be either edge or level sensitive. Bit 1 = 1 defines LRQnN to be edge sensitive.

Control Register 'n' Bit 2

Selects either vectored mode or device-supplies-the-vector mode response. Bit 2 = 1 enables vectored mode operation for LRQnN.

Two modes of operation for a local interrupt response are possible; vectored mode and device-supplies-the-vector mode. In vectored mode, the SCB68155/8X824 supplies the interrupt vector to the local CPU and asserts LDTACKN to complete the transfer. In the device-supplies-the-vector mode, the local interrupting device supplies its own interrupt vector and asserts LDTACKN to complete the transfer.

The vector register R2 allows the user to program the five most significant bits (bits 7 – 3) of the interrupt vector supplied in vectored mode. During a vectored local interrupt acknowledge cycle, the upper five bits of the vector register are concatenated with a 3-bit interrupt level (address lines A3 = B2 of the vector, A2 = B1 and A1 = B0). This forms the unique vector for the local interrupt request level being acknowledged.

The local interrupt request mask register R3 allows the user to selectively enable local interrupt requests by setting appropriate bits in the register.

The current state of the local interrupt requests can be determined by the local master by reading the local interrupt status register R4.

Local Interrupt Acknowledge

An interrupt acknowledge, by the local CPU, is signified by the assertion of the interrupt acknowledge input (IACKDSN). The SCB68155/8X824 responds by reading the three address lines (A1 – A3) to determine what level is being acknowledged. If a local interrupt is the highest priority interrupt pending on the level acknowledged, the SCB68155/8X824 will respond as though it is programmed for that level.

If vectored mode is programmed, the SCB68155/8X824 will assert the local interrupt acknowledge (LIACKN) and place the interrupt vector on the local data bus. To complete the transfer of the vector to the local CPU, the SCB68155/8X824 asserts the local data transfer acknowledge signal (LDTACKN).

If device-supplies-the-vector mode is programmed, the SCB68155/8X824 asserts the local interrupt acknowledge signal (LIACKN). The interrupting device is then allowed to place its own vector on the local data bus and assert LDTACKN.

When a local interrupt is acknowledged by the local master, the appropriate bit in the LRQ status register R4 is cleared to 0.

Bus Interrupts

The VMEbus specification defines a maximum of seven interrupt levels. The SCB68155/8X824 can handle seven system bus interrupts through its IRQ1N – IRQ7N lines. Bus interrupt request are active low level sensitive, and prioritized with IRQ7N being the highest priority and IRQ1N the lowest priority. The bus mask control register R5 allows the user to selectively enable bus interrupt requests by setting appropriate bits in the register. The local CPU can read the bus interrupt status register R6 to determine the current state of the bus interrupt requests.

Bus Interrupt Acknowledge

The local CPU asserts the interrupt acknowledge signal (IACKDSN) to signify an interrupt acknowledge. The SCB68155/8X824 reponds by reading the interrupt level on A1 – A3 to determine what level is being acknowledged. If a local interrupt is not pending on the level acknowledged, and that bus level is not masked, the SCB68155/8X824 will assert bus interrupt acknowledge (BIACKN). If that bus level is masked, the SCB68155/8X824 will not respond to the interrupt acknowledge by the local master.

Part of the interrupt acknowledge sequence for a bus interrupt consists of acquiring a vector (status/ID byte) from the system bus.

Table 1. SCB68155/8X824 Interrupt Level Encoding

INTERRUPT REQUEST LEVEL	INTERRUPT PRIORITY LEVEL OUTPUTS		
	IPL2N	IPL1N	IPL0N
NMIN, IRQ7N	0	0	0
LRQ6N, IRQ6N	0	0	1
LRQ5N, IRQ5N	0	1	0
LRQ4N, IRQ4N	0	1	1
LRQ3N, IRQ3N	1	0	0
LRQ2N, IRQ2N	1	0	1
LRQ1N, IRQ1N	1	1	0
None	1	1	1

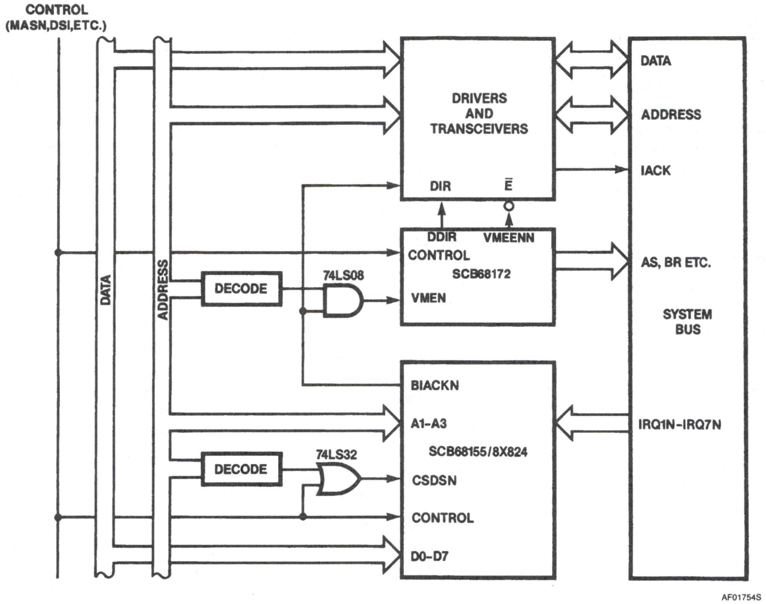

CONTROL
(MASN,DSI,ETC.)
DATA
ADDRESS
DRIVERS
AND
TRANSCEIVERS
DIR
Ē
DATA
ADDRESS
IACK
DDIR
VMEENN
CONTROL
SCB68172
VMEN
AS, BR ETC.
SYSTEM
BUS
DECODE
74LS08
BIACKN
A1-A3
SCB68155/8X824
CSDSN
CONTROL
D0-D7
IRQ1N-IRQ7N
DECODE
74LS32
AF01754S

Figure 1. SCB68155/8X824-SCB68175/8X821 Typical System Configuration

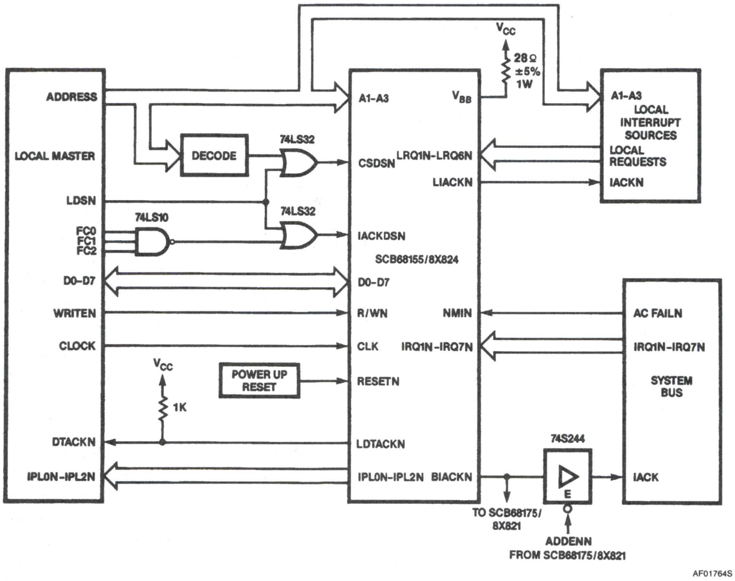

ADDRESS
LOCAL MASTER
LDSN
FC0
FC1
FC2
D0-D7
WRITEN
CLOCK
DTACKN
IPL0N-IPL2N
DECODE
74LS32
74LS10
74LS32
VCC
1K
POWER UP
RESET
A1-A3
VBB
CSDSN
LRQ1N-LRQ6N
LIACKN
IACKDSN
SCB68155/8X824
D0-D7
R/WN
NMIN
CLK
IRQ1N-IRQ7N
RESETN
LDTACKN
IPL0N-IPL2N
BIACKN
VCC
28Ω
±5%
1W
A1-A3
LOCAL
INTERRUPT
SOURCES
LOCAL
REQUESTS
IACKN
AC FAILN
IRQ1N-IRQ7N
SYSTEM
BUS
74S244
E
IACK
TO SCB68175/
8X821
ADDENN
FROM SCB68175/8X821
AF01764S

Figure 2. Typical SCB68155/8X824 Local Master Configuration

The bus signals required to acquire this vector are available with a bus controller. The Signetics SCB68172 bus controller can be used by the SCB68155/8X824 to acquire the vector (status/ID byte), thereby eliminating the need for the SCB68155/8X824 to duplicate this bus control function. Because most interrupts are serviced by boards that already have the SCB68172, a one-chip addition of the SCB68155/8X824 gives that board complete interrupt handling capability.

Since the SCB68155/8X824 is an asynchronous device, it is possible for a local interrupt request to be asserted during acknowledgement of a bus interrupt on the same level. The SCB68155/8X824 passes all local interrupt requests through transparent latches which close during each interrupt acknowledge cycle. All possibility of contention is therefore eliminated.

Reset

When RESETN is asserted, the SCB68155/8X824 drives LDTACKN, LIACKN, BIACKN and IPL0N – IPL2N all high. The D0 – D7 I/O pins go to 3-State and all internal registers are cleared.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	–65 to +150	°C
V_{CC}	Supply voltage[3] range	–0.5 to +7.0	V
V_{IN}	Input voltage[3] range	–0.5 to +5.5	V
V_{OUT}	Voltage applied to output in off-state[3]	–0.5 to +5.5	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS Min	LIMITS Max	UNIT
V_{CC}	Supply voltage		4.75	5.25	V
V_{BB}	Supply voltage		1.35	1.65	V
I_{CC}	V_{CC} supply current	$V_{CC} = 5.25V$		65	mA
I_{BB}	V_{BB} supply current	$V_{BB} = 1.65V$		190	mA
I_{IL}	Input low current	$V_{CC} = 5.25V$, $V_{BB} = 1.65V$, $V_{IL} = 0.4V$		–20	μA
I_{IH}	Input high current	$V_{CC} = 5.25V$, $V_{BB} = 1.65V$, $V_{IH} = 2.7V$		20	μA
I_{OS}	Short circuit output current except LDTACKN	$V_{CC} = 5.25V$, $V_{OUT} = 0V$[6]	–15	–100	mA
V_{OL}	Output low voltage	$V_{CC} = 4.75V$, $V_{BB} = 1.35V$, $I_{OL} = 8mA$		0.6	V
V_{OH}	Output high voltage except LDTACKN	$V_{CC} = 4.75V$, $V_{BB} = 1.35V$, $I_{OH} = -3mA$	2.5		V
I_I	Input leakage current	$V_{CC} = 5.25V$, $V_{IN} = 5.25V$		100	μA
I_{CEX}	Open collector leakage current LDTACKN	$V_{CC} = 4.75V$, $V_{OUT} = 4.25V$		100	μA
V_{IC}	Input clamp voltage	$V_{CC} = 4.75V$, $I_{IN} = -10mA$	–1.5		V
V_{IL}	Input low voltage			0.8	V
V_{IH}	Input high voltage		2		V

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient for ceramic package (116°C/W for plastic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all inputs swing between 0.4V and 2.4V with a transition time of 3ns maximum and output voltages are checked at 0.8V and 2V.
6. At any time, no more than one output should be connected to ground.
7. t_{TST} is always greater than or equal to t_{DTH}.
8. t_{TRST} is always greater than or equal to t_{DAH}.

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5V ±5%[4, 5]

SYMBOL	PARAMETER	TENTATIVE LIMITS		UNIT
		Min	Max	
Register read (see Figure 3)				
t_{RWS}	R/WN to CSDSN low setup time	10		ns
t_{IAKS}	IACKDSN high to CSDSN low setup time	10		ns
t_{ADRS}	A1 – A3 valid to CSDSN low setup time	30		ns
t_{DTV}	CSDSN low to D0 – D7 setup time		89	ns
t_{ACCR}	CSDSN low to LDTACKN low read access time		$2t_{CKPD}$ + 116[8]	ns
t_{RWH}	CSDSN high to R/WN high hold time	0		ns
t_{ADRH}	CSDSN high to A1 – A3 valid hold time	0		ns
t_{DTH}	CSDSN high to D0 – D7 valid hold time	0	79	ns
t_{TST}[7]	CSDSN high to D0 – D7 3-State	0	80	ns
t_{ACK}	CSDSN high to LDTACKN high time	0	66	ns
t_{CSH}	CSDSN high time	10		ns
t_{DTCS}	LDTACKN low to CSDSN high	0		ns
Register write (see Figure 4)				
t_{RWS2}	R/WN low to CSDSN low setup time	10		ns
t_{IAKS}	IACKDSN high to CSDSN low setup time	10		ns
t_{ADRS}	A1 – A3 valid to CSDSN low setup time	30		ns
t_{DS}	D0 – D7 valid to CSDSN low setup time	0		ns
t_{ACCW}	CSDSN low to LDTACKN low write access time		$2t_{CKPD}$ + 116[8]	ns
t_{RWH2}	CSDSN high to R/WN low hold time	0		ns
t_{ADRH}	CSDSN high to A1 – A3 valid hold time	0		ns
t_{DH}	CSDSN high to D0 – D7 valid hold time	0		ns
t_{ACK}	CSDSN high to LDTACKN high time	0	66	ns
t_{CSH}	CSDSN high time	100		ns
t_{DTCS}	LDTACKN low to CSDSN high time	0		ns
Vector mode (see Figure 5)				
t_{CSS}	CSDSN high to IACKDSN low setup time	10		ns
t_{PDL}	IACKDSN low to LIACKN low propagation time	t_{CKPD}	$2t_{CKPD}$ + 68	ns
t_{DAV}	IACKDSN low to D0 – D7 vector valid		103	ns
t_{ACCV}	IACKDSN low to LDTACKN low (vector access time)	t_{CKPD}	t_{CKPD} + 116[8]	ns
t_{IKH}	IACKDSN high time	100		ns
t_{DAH}	IACKDSN high to D0 – D7 valid hold time	0	111	ns
t_{TRST}[8]	IACKDSN high to D0 – D7 3-State	0	115	ns
t_{IKDT}	IACKDSN high to LDTACKN high	0	81	ns
t_{PDH}	IACKDSN high to LIACKN high propagation delay	0	42	ns
t_{DTIK}	LDTACKN low to IACKDSN high time	0		ns
t_{ADRS}	A1 – A3 valid to IACKDSN low setup time	0		ns
t_{ADH}	IACKDSN high to A1 – A3 valid hold time	0		ns
Device supplies the vector mode (see Figure 6)				
t_{CSS}	CSDSN high to IACKDSN low setup time	10		ns

AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	TENTATIVE LIMITS		UNIT
		Min	Max	
t_{PDL}	IACKDSN low to LIACKN low propagation time delay	t_{CKPD}	$2t_{CKPD} + 68$	ns
t_{IKH}	IACKDSN high time	100		ns
t_{PDH}	IACKDSN high to LIACKN high propagation delay	0	42	ns
t_{ADS}	A1 – A3 valid to IACKDSN low setup time	0		ns
t_{ADH}	IACKDSN high to A1 – A3 valid hold time	0		ns
Bus interrupt acknowledge (see Figure 7)				
t_{CSS}	CSDSN high IACKDSN low setup time	10		ns
t_{PDL2}	IACKDSN low to BIACK low propagation delay	t_{CKPD}	$2t_{CKPD} + 68$	ns
t_{IKH}	IACKDSN high time	100		ns
t_{PDH2}	IACKDSN high to BIACK high propagation delay	0	50	ns
t_{ADS}	A1 – A3 valid to IACKDSN low setup time	0		ns
t_{ADH}	IACKDSN high to A1 – A3 hold time	0		ns
Reset timing (see Figure 8)				
t_{RST}	RESET low time	120		ns
Clock timing (see Figure 9)				
t_{CKPD}	Clock period	100		ns
t_{CKH}	Clock high	50		ns

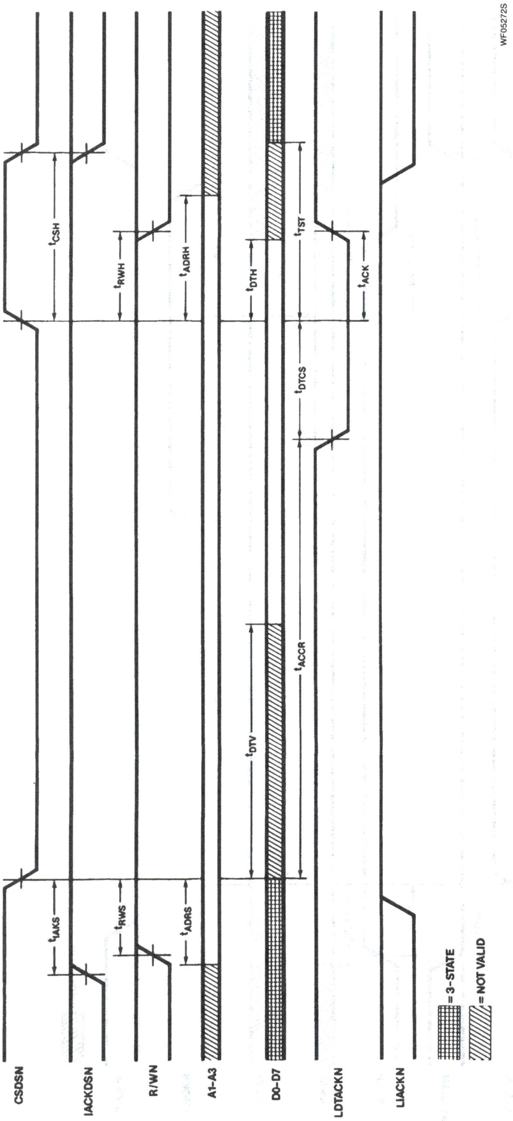

CSDSN
IACKDSN
R/WN
A1–A3
D0–D7
LDTACKN
LIACKN
t_{CSH}
t_{RWH}
t_{ADRH}
t_{DTH}
t_{TST}
t_{ACK}
t_{DTCS}
t_{ACCR}
t_{DTV}
t_{IAKS}
t_{RWS}
t_{ADRS}
= 3-STATE
= NOT VALID
WF05272S

Figure 3. Register Read

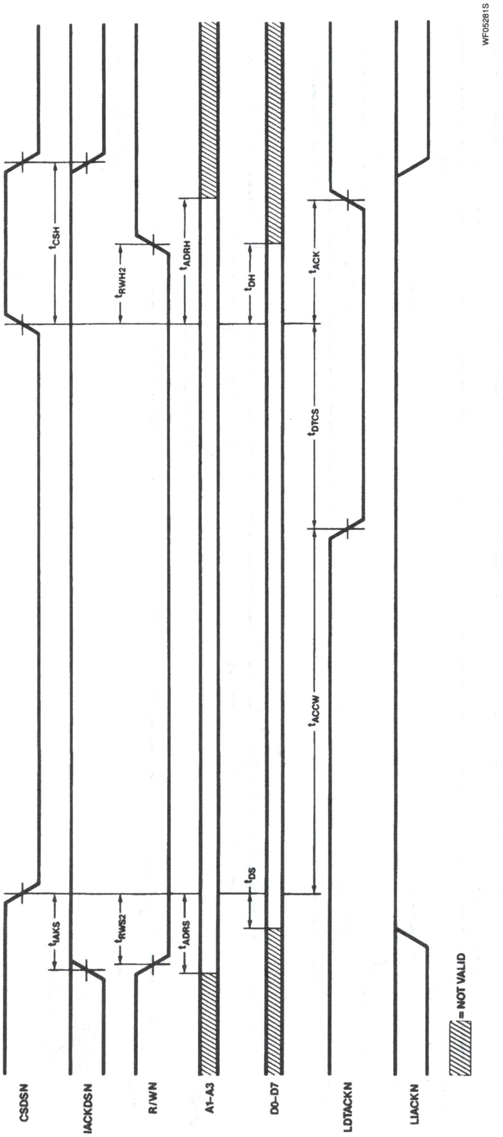

CSDSN
IACKDSN
R/WN
A1–A3
D0–D7
LDTACKN
LIACKN
t_{CSH}
t_{RWH2}
t_{ADRH}
t_{DH}
t_{ACK}
t_{DTCS}
t_{ACCW}
t_{AKS}
t_{RWS2}
t_{ADRS}
t_{DS}
▨ = NOT VALID
WF05281S

Figure 4. Register Write

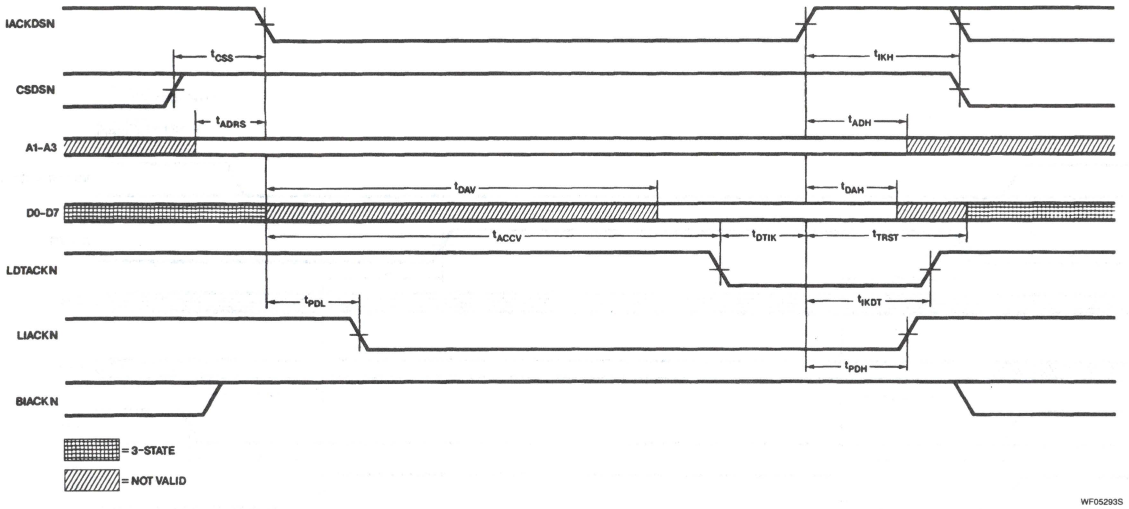

IACKDSN
CSDSN
A1-A3
D0-D7
LDTACKN
LIACKN
BIACKN
t_{CSS}
t_{ADRS}
t_{DAV}
t_{ACCV}
t_{PDL}
t_{DTIK}
t_{IKH}
t_{ADH}
t_{DAH}
t_{TRST}
t_{IKDT}
t_{PDH}
▦ = 3-STATE
▨ = NOT VALID
WF05293S

Figure 5. Local Interrupt Acknowledge (Vector Mode)

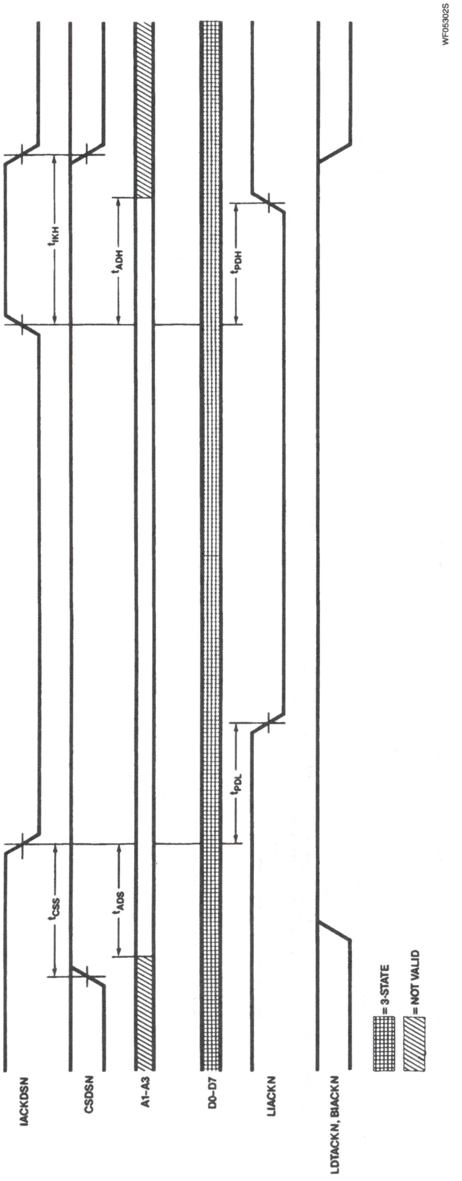

IACKDSN
CSDSN
A1–A3
D0–D7
LIACKN
LDTACKN, BIACKN
t_{IKH}
t_{ADH}
t_{PDH}
t_{PDL}
t_{CSS}
t_{ADS}
= 3-STATE
= NOT VALID
WF05302S

Figure 6. Local Interrupt Acknowledge (Non-Vectored Mode)

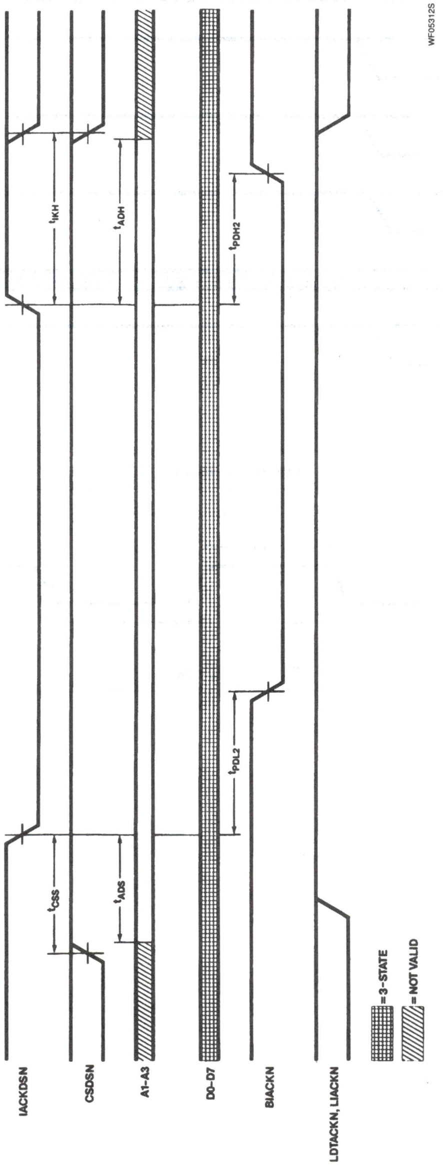

IACKDSN
CSDSN
A1–A3
D0–D7
BIACKN
LDTACKN, LIACKN
tIKH
tADH
tPDH2
tCSS
tADS
tPDL2
=3-STATE
=NOT VALID
WF05312S

Figure 7. Bus Interrupt Acknowledge

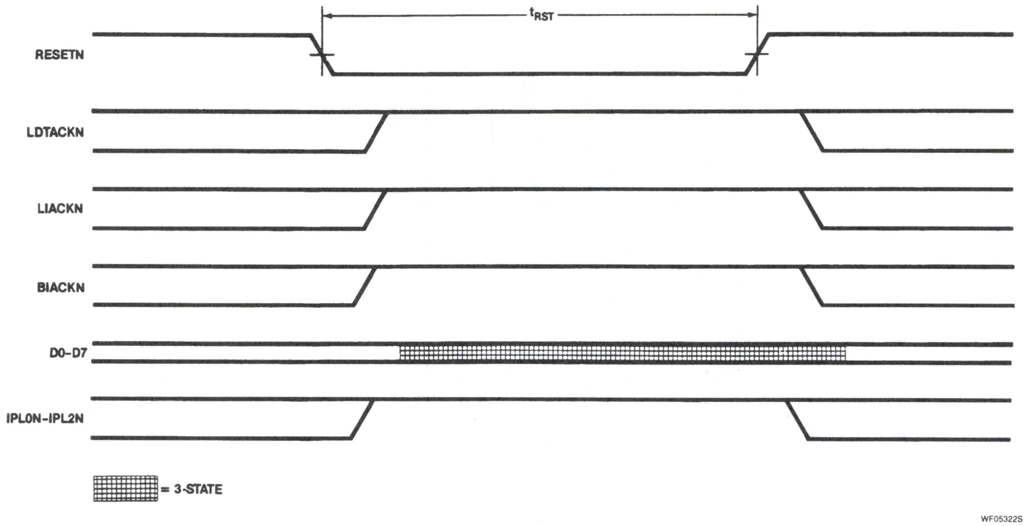

tRST
RESETN
LDTACKN
LIACKN
BIACKN
D0-D7
IPL0N-IPL2N
= 3-STATE
WF05322S

Figure 8. Reset

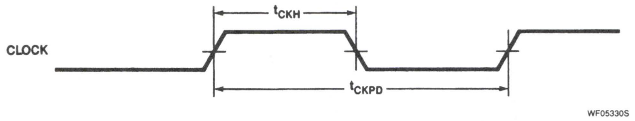

tCKH
CLOCK
tCKPD
WF05330S

Figure 9. Clock

Signetics

Microprocessor Products

SCB68171
Very Little Serial Interface Chip (VLSIC)

Preliminary Specification

DESCRIPTION

The Signetics Very Little Serial Interface Chip (VLSIC) is a bipolar interface device which connects one or more VMSbus controllers to the VMSbus itself. It provides bus driving and receiving in addition to latching data in both the transmit and receive directions.

SERCLK on the VMSbus has a waveform as shown in Figure 1, with four edges per cycle which are designated C1, S1, C2, S2. SYSCLK is used to discriminate (differentiate) the phases of SERCLK. The SYSCLK input should have a nominal 50% duty cycle and a cycle time which is 2/11 that of SERCLK, but SYSCLK and SERCLK need not be synchronous or have any fixed phase relationship. The 16MHz SYSCLK on the VMEbus meets these requirements for a back plane 2.9MHz SERCLK.

The VLSIC samples the XDATAN input on the SERCLK edge designated C1. XDATAN is clocked directly to SERDATN. The VLSIC then clocks the low-active OR (positive logic AND) of SERDATN and SERDATIN to RDATAN on the S1 edge. If XDATAN was high at C1, the VLSIC then samples the XSTARTN input on the C2 edge, and if it is low, the VLSIC makes SERDATN low, thus making a VMSbus start bit. If XDATAN was high at S1, then on the S2 edge it clocks the low active OR of SERDATN and SERDATIN to RSTARTN. If SERDATN was low at S1, it keeps (or makes) RSTARTN high at S2.

CHIPCLK is driven high from the C1 edge of SERCLK, and driven low from the C2 edge. Thus RDATAN and RSTARTN setup to CHIPCLK edges by approximately the low time of SERCLK. The VMSbus controller(s) must meet the specified setup and hold times to C1 for XDATAN and to C2 for XSTARTN.

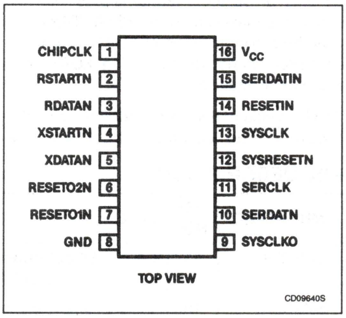
FEATURES

- **70mA open-collector drive for SERDATN**
- **Low capacitive loading**
- **Discriminates SERCLK into single-phase chip clock output**
- **Separates data and start bits for both receive and transmit**
- **Provides single bus load for multiple VMSbus controllers**
- **Simplifies controller design and allows use of slower technology**
- **VMEbus receiver for SYSCLK**
- **VMEbus driver/receiver for SYSRESETN**
- **16-pin DIP**

ORDERING INFORMATION

DESCRIPTION	PARAMETER	ORDERING CODE
16-Pin Plastic DIP	V_{CC} 5V ±5%, $T_A = 0$ to 70°C	SCB68171C3N16

PIN CONFIGURATION



Very Little Serial Interface Chip (VLSIC) SCB68171

DESCRIPTION (Continued)

A VMSbus controller is required to "jam" the bus by sending a string of "ones" (low) on SERDATN when a start bit is sensed while the controller is sending or tracking a frame. The first one bit should directly follow the misplaced start bit. Since the minimum S2-to-C1 time of SERCLK (25ns) is less than the sum of the maximum S2-to-RSTART-low time plus the minimum setup of XDATAN to C1, a controller cannot do this using XDATAN in the normal fashion.

The following feature is provided to solve this problem. The condition XDATAN low, XSTARTN low, and CHIPCLK high directly sets the flipflop controlling SERDATN, and makes SERDATN low. (Note that a VMSbus controller would never assert both XDATAN and XSTARTN low in normal operation.) The assertion of both XDATAN and XSTARTN must occur soon enough to satisfy the SERDATN to S1 setup requirements of all the modules on the VMSbus.

The VLSIC is primarily intended for use in the P1 region of a VMEbus card. Space and functionality is at a premium in this area. Accordingly, the VLSIC includes an autonomous function of driving and receiving SYSRESETN on the VMEbus. A low on the RESETIN input makes the VLSIC drive the SYSRESETN pin low. SYSRESETN is also received and driven onto two open-collector outputs RESETO1N and RESETO2N. RESETO1N has a high drive capability and is suitable for connection to the RESETN pin of a 680x0 processor, while RESETO2N has lower drive and capacitance and can be connected to the processor's HALTN pin. This function has no connection to the rest of the VLSIC, and could thus be used for some other purpose.

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
SERCLK	11	I	**Direct connect to VMSbus clock:** Clocks SERDATN to RDATAN, RSTARTN. Clocks XDATAN, XSTARTN to SERDATN. Used to generate single-phase CHIPCLK.
SYSCLK	13	I	**Direct connect to VMEbus clock or other signal meeting the specified frequency relationship to SERCLK:** Used to discriminate the phases of SERCLK.
SYSCLKO	9	O	**Repeater of SYSCLK for onboard use.**
SERDATN	10	I/O	**Direct connect to VMSbus data in bipolar open-collector links:** Provides data output function for other types of links (e.g. optical). Requires pull-up resistor in either case.
SERDATIN	15	I	**Tied to high logic level in bipolar open-collector links:** Provides data input function for other types of links (e.g. optical).
CHIPCLK	1	O	**Single phase clock for VMSbus controllers.**
RDATAN	3	O	**Conveys one/zero bits to VMSbus controller(s).**
RSTARTN	2	O	**Conveys start bits to VMSbus controller(s).**
XDATAN	5	I	**Input from VMSbus controller(s):** In normal operation, a low on this line indicates a "one" bit should be sent. Simultaneous assertion of XDATAN and XSTARTN low, while CHIPCLK is high, drives SERDATN low directly in a "jam" condition.
XSTARTN	4	I	**Input from VMSbus controller(s):** In normal operation, a low on this line indicates a "start" bit should be sent. Simultaneous assertion of XDATAN and XSTARTN low, while CHIPCLK is high, drives SERDATN low directly in a "jam" condition.
RESETIN	14	I	**Input from onboard logic:** Low state forces SYSRESETN low.
SYSRESETN	12	I/O	**Direct connect to VMEbus system reset:** Open-collector output from RESETIN, received to drive RESETO1N and RESETO2N. Does not affect other VLSIC logic.
RESETO1N	7	O	**High-drive open-collector output from SYSRESETN.**
RESETO2N	6	O	**Low-drive open-collector output from SYSRESETN.**
V_{CC}	16		**+5 Volts**
GND	8		**Ground**

VMSBUS CONTROLLER DESIGN

Controllers using VLSIC should signal as follows on XDATAN and XSTARTN.

1. Controllers should present the next bit on XDATAN in response to the falling edge of CHIPCLK, and on XSTARTN in response to the rising edge of CHIPCLK. For a 2.9MHz SERCLK, they have at least 90nsec to do so, and approximately 120nsec from RDATAN valid.
2. Controllers may release XDATAN to high in response to the rising edge of CHIPCLK, and may release XSTARTN to high in response to the falling edge. Since these are typically open-collector outputs of the controllers, there may be a timing advantage to do so.
3. A controller should present XDATAN and XSTARTN low in a "jam" situation, in a combinatorial fashion from RSTARTN. Thereafter the controller can release XSTARTN from the lowgoing edge of CHIPCLK, and may either signal 511 or 512 one bits in the usual fashion (1 and 2 above), or may just keep XDATAN low.
4. If two VMSbus controllers connected to the same VLSIC become "locally desynchronized", it is possible that one will present XDATAN low and the other XSTARTN low for the same bit cell. If this occurs, XDATAN predominates and SERDATN is driven low for a "one" bit. Thus a possible transient combination of XDATAN and XSTARTN low and CHIPCLK high actually has no effect. The controller presenting XSTARTN thereafter receives RDATAN low, and continues to try to send the start bit.

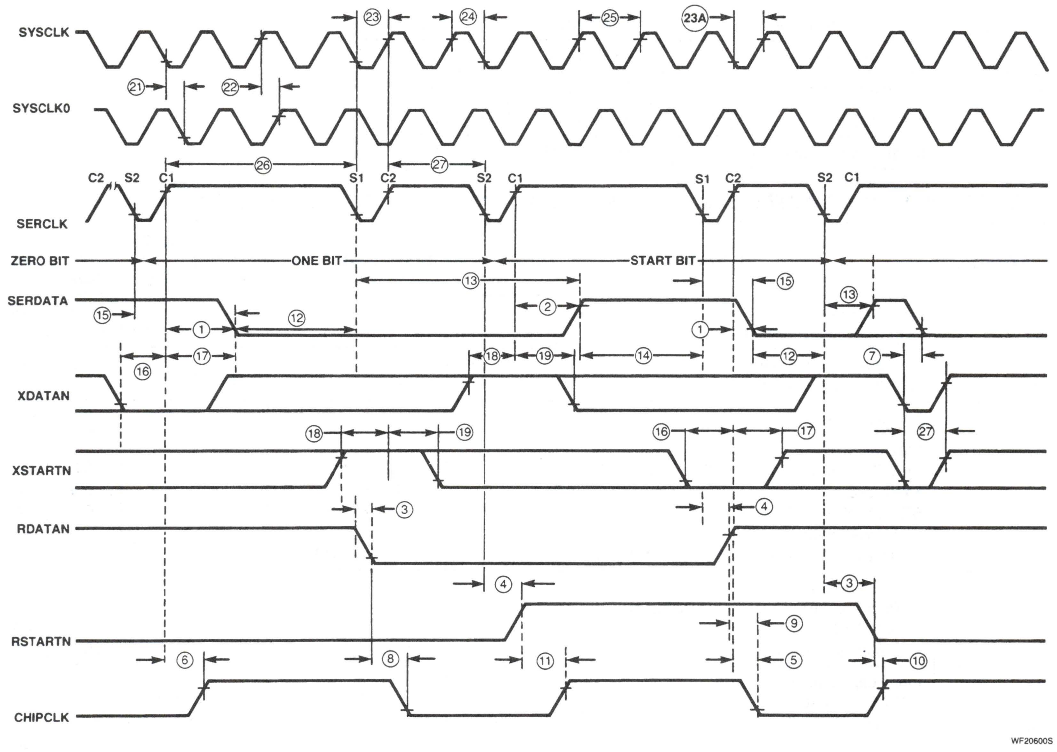

SYSCLK
SYSCLK0
SERCLK
ZERO BIT
ONE BIT
START BIT
SERDATA
XDATAN
XSTARTN
RDATAN
RSTARTN
CHIPCLK
C2
S2
C1
S1
23A
WF20600S

Figure 1. VLSIC Timing

Very Little Serial Interface Chip (VLSIC) SCB68171

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
V_S	Supply voltage range	−0.5 to +7	V
V_{IN}	Input voltage range	−0.5 to +5.5	V
T_A	Operating temperature range[2]	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to $+70°C$, $R_L = 90$, $C_L = 15pF$[3, 4]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
I_{IL}	Low-level input current	$V_{IN} = 0.4V$			−0.4	mA
I_{IH}	High-level input current	$V_{IN} = 2.7V$			20	µA
V_{IL}	Low-level input voltage				0.8	V
V_{IH}	High-level input voltage		2			V
$V_{TH+} - V_{TH-}$	Hysteresis — All inputs			300		mV
V_{OL}	Low level output voltage					
	CHIPCLK, RDATAN, RSTARTN, RESETO2N	$I_{OL} = 8mA$			0.5	V
	SYSCLKO	$I_{OL} = 24mA$			0.5	V
	SERDATN, RESETO1N, SYSRESETN	$I_{OL} = 70mA$			0.5	V
V_{OH}	High level output voltage					
	CHIPCLK, RDATAN, RSTARTN, SYSCLKO	$I_{OH} = -3mA$	2.7			V
I_{OH}	Output leakage current					
	SERDATN, RESETO1N, SYSRESETN					
	RESETO2N	$V_{OH} = 5.5V$			100	µA
I_{CC}	V_{CC} supply current	$V_{CC} = Max$			60	mA
C_I	Input capacitance					
	SERCLK, SYSCLK			5		pF
C_{IO}	I/O capacitance					
	SERDATN			10		pF

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other conditions other than those indicated in the Electrical Characteristics section of this data sheet is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. Parameters are valid over specified temperature range.
4. All voltage measurements are referenced to ground (GND). All time measurements are referenced at input voltages of 0.8V and 2V as appropriate.

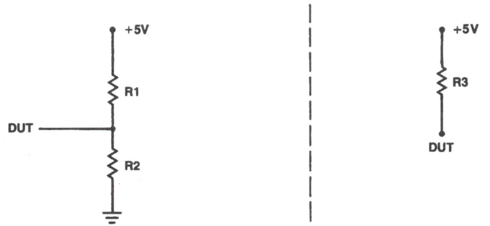
Very Little Serial Interface Chip (VLSIC) SCB68171

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to $+70°C$, $R_L = 90$, $C_L = 50pF^{3,\ 4}$ (except as noted)

NO.	CHARACTERISTIC	TENTATIVE LIMITS		UNIT
		Min	Max	
1	Prop, C1 or C2 to SERDATN low	10	20	ns
2	Prop, C1 to SERDATN released	10	25	ns
3	Prop, S1 (S2) to RDATAN (RSTARTN) low		20	ns
4	Prop, S1 (S2) to RDATAN (RSTARTN) high		20	ns
5	Prop, C2 to CHIPCLK low		15	ns
6	Prop, C1 to CHIPCLK high		12	ns
7	Prop, XDATAN low and XSTARTN low (with CHIPCLK high) to SERDATN low[5]		25	ns
8	RDATAN low to CHIPCLK low	t_{CL-10}		ns
9	RDATAN high to CHIPCLK low	t_{CL-12}		ns
10	RSTARTN low to CHIPCLK high	t_{CL-10}		ns
11	RSTARTN high to CHIPCLK high	t_{CL-12}		ns
12	Setup, SERDATN and/or SERDATIN low to S1 or S2	8		ns
13	Hold, SERDATN and/or SERDATIN low after S1 or S2	2		ns
14	Setup, SERDATN and SERDATIN high to S1 or S2	6		ns
15	Hold, SERDATN and SERDATIN high after S1 or S2	2		ns
16	Setup, XDATAN low to C1, XSTARTN low to C2	7		ns
17	Hold, XDATAN low after C1, XSTARTN low after C2	0		ns
18	Setup, XDATAN high to C1, XSTARTN high to C2	7		ns
19	Hold, XDATAN high after C1, XSTARTN high after C2	0		ns
20	Pulse Width, XDATAN and XSTARTN low with CHIPCLK high	10		ns
21	Prop, SYSCLK low to SYSCLKO low	8	15	ns
22	Prop, SYSCLK high to SYSCLKO high	5	12	ns
23	Pulse width, SERCLK low (t_{CL})	15		ns
23A	Pulse width, SYSCLK low (t_{SYCL})	10		ns
24	Pulse width, SYSCLK high (t_{SYCH})	10		ns
25	Cycle time, SYSCLK (t_{SYCY})	62		ns
25A	Cycle time, SERCLK (general case)	5.49(t_{SYSY})	551(t_{SYSY})	ns
26	Pulse width, SERCLK high, C1 to S1			
	($t_{SYCY} = 62.5$)	174.25		ns
	(general case)	$2.5t_{SYCY} + 18$		ns
27	Pulse width, SERCLK high, C2 to S2			
	($t_{SYCY} = 62.5$)		120	ns
	(general case)		2(t_{SYCY}) − 5	ns

TEST CONDITIONS

TEST CONDITIONS:



SIGNALS	R1	R2
CHIPCLK, RSTARTN, RDATAN	453Ω	267Ω
SYSCLKO	162Ω	124Ω

SIGNALS	R3
SYSRESETN, RESET01N, SERDATN	69.8Ω
RESET02N	562Ω

TC21130S

Signetics

SCB68172
VMEbus Controller (BUSCON)

Preliminary Specification

Microprocessor Products

DESCRIPTION

The Signetics SCB68172 VMEbus Controller (BUSCON) is an interface device for the VMEbus. It can be used in three different configurations: master-only, slave-only, and master/slave. The SCB68172 can be used with a processor-type interface or with a DMA controller-type interface. In all configurations, it handles the VMEbus signaling protocol in compliance with revisions B and C of the VMEbus Specification.

CONFIGURATION/VERSION

Applications of the BUSCON are identified as follows (see Figures 1 through 4):

VERSION	APPLICATION
PMS	Processor-type master/slave
DMAC	DMA controller-type master/slave
MS	Either PMS or (DMAC)
M	Master-only
S	Slave-only

All of these applications are handled by the SCB68172, with unused pins tied to stated logic levels in some of the applications.

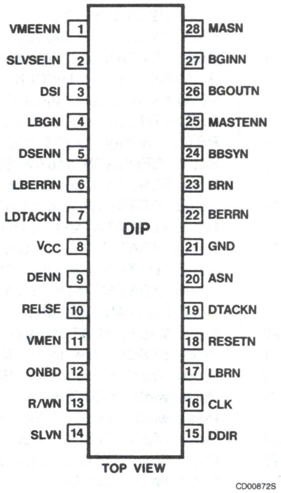
Figure 5 shows a functional model of the SCB68172 logic. The ASN, MASN, LBRN, BGINN, and RELSE inputs are internally synchronized to CLK before being presented to the state machine which determines the major functions of the device. The SLVN, ONBD, and VMEN signals are used directly in the state machine, although they are highly qualified to prevent metastable conditions on the state machine outputs. The BRN, BBSYN and LBGN signals are direct state machine outputs, while ASN, MASTENN, VMEENN, SLVSELN, and BGOUTN are derived from the state machine outputs plus some combinatorial qualification.

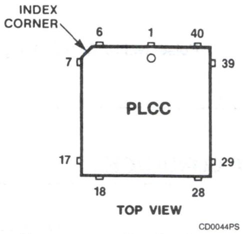
The DSI, R/WN, DTACKN, BERRN, LDTACKN, and LBERRN inputs function largely as direct combinatorial inputs. The DDIR, DTACKN, BERRN, LDTACKN, LBERRN, and (when applicable) MASN outputs are largely derived directly from these direct inputs, with some qualification from the state machine outputs. The DENN and DSENN outputs have complex multi-case logic which uses both the direct inputs and the state machine outputs.

FEATURES

- **Master, slave, or master/slave (dual ported) applications**
- **Helps assure VMEbus compatibility**
- **Allows for address decoding time**
- **Processor or DMA controller interface for master/requester**
- **Master/requester logic allows release on request (ROR) or release when done (RWD) operation, early or intercycle release**
- **Supports and exploits address lookahead**

PIN CONFIGURATIONS


VMEENN 1
SLVSELN 2
DSI 3
LBGN 4
DSENN 5
LBERRN 6
LDTACKN 7
Vcc 8
DENN 9
RELSE 10
VMEN 11
ONBD 12
R/WN 13
SLVN 14
28 MASN
27 BGINN
26 BGOUTN
25 MASTENN
24 BBSYN
23 BRN
22 BERRN
21 GND
20 ASN
19 DTACKN
18 RESETN
17 LBRN
16 CLK
15 DDIR
DIP
TOP VIEW
CD00872S


INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

Pin	Function	Pin	Function
1	NC	23	NC
2	VMEENN	24	DDIR
3	SLVSELN	25	CLK
4	DSI	26	LBRN
5	LBGN	27	RESETN
6	NC	28	NC
7	NC	29	NC
8	DSENN	30	DTACKN
9	LBERRN	31	ASN
10	NC	32	NC
11	LDTACKN	33	GND
12	NC	34	NC
13	V_{CC}	35	BERRN
14	NC	36	NC
15	DENN	37	BRN
16	RELSE	38	BBSYN
17	NC	39	NC
18	NC	40	NC
19	VMEN	41	MASTENN
20	ONBD	42	BGOUTN
21	R/WN	43	BGINN
22	SLVN	44	MASN

VMEbus Controller (BUSCON) SCB68172

ORDERING INFORMATION

PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to 70°C
CERDIP	SCB68172C2F28
Plastic DIP	SCB68172C2N28
Plastic LCC	SCB68172C2A44

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	CONFIG	NAME AND FUNCTION
	DIP	PLCC			
CLK	16	25	I	All	**Clock:** User-supplied clock signal.
SLVN	14	22	I	S,MS	**Slave:** Active-low decode of the VMEbus address and address modifier lines indicating that the current cycle is for this board. SLVN should not be qualified with ASN nor VMEENN. It is first sampled on the rising clock edge after the rising edge on which ASN is first detected. It must remain valid until after the next low-going edge on DTACKN or BERRN. In a master-only application, SLVN should be pulled up to V_{CC}.
ASN	20	31	I/O I	M,MS S	**Address Strobe:** Direct connect to VMEbus ASN.
VMEN	11	19	I	M,MS	**VME Decode:** Active-low decode of the master's address lines, indicating that the master's current cycle is for a slave on the VMEbus. VMEN should not be qualified with MASN nor MASTENN. It is first sampled on the rising clock edge after the one on which MASN is first detected. Thereafter, it must remain valid until MASN goes false (high). In a slave-only configuration, VMEN should be pulled up to V_{CC}.
LBRN	17	26	I	M,MS	**Local Bus Request:** Connected to the low-active bus request output of a DMA controller. Typically tied to a high logic level in processor-type interfaces.
ONBD	12	20	I	MS	**Onboard:** Active-high decode of the master's address lines, indicating that the master's current cycle is for an onboard slave that is dual-ported with the VMEbus. ONBD should not be qualified with MASN or MASTENN. It is first sampled on the rising clock edge after the one on which MASN is first detected. Thereafter, it must remain valid until after MASN goes false (high). In a master-only or slave-only application, ONBD should be grounded. If a master/slave configuration does not contain "local slaves" as shown in Figure 3, VMEN and ONBD should both be connected to an active-low "VME decode". A cycle between the onboard master and a local slave (VMEN high, ONBD low) is ignored by BUSCON, and can proceed concurrently with a cycle between another VMEbus master and an onboard dual-ported slave.
MASN	28	44	I I/O	M,PMS DMAC	**Master's Address Strobe:** RMW and Sequential VMEbus master cycles are accomplished by holding MASN low across several data strobes. If LBGN is high at the end of the RESETN low time, the state of ASN is driven onto MASN whenever BUSCON does not have control of the VMEbus. In a slave-only application, MASN should be pulled up to V_{CC}.
MASTENN	25	41	O	MS	**Master Enable:** In a master/slave application, the low state of this signal enables the master onto the shared bus and enables shared-bus responses back to the master. MASTENN also provides the direction control for the VMEbus address transceivers.
VMEENN	1	2	O	M,MS	**VME Enable:** Active-low enable for the VMEbus address drivers (master-only) or transceivers (master/slave).
SLVSELN	2	3	O	S,MS	**Slave Select:** Active-low select for the onboard slave resources (the shared/dual ported slaves in a master/slave application). Derived from MASN and ONBD, or from ASN and SLVN. If necessary, MASTENN and VMEENN are cycled to provide address setup time before SLVSELN is asserted.
BRN	23	37	O	M,MS	**Bus Request:** Active-low, open collector VMEbus request. Direct connect to the selected level among VMEbus BR0* – BR3*.
BGINN	27	43	I	M,MS	**Bus Grant In:** Direct connect to the selected level among VMEbus BG0IN* – BG3IN*.

VMEbus Controller (BUSCON) SCB68172

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	CONFIG	NAME AND FUNCTION
	DIP	PLCC			
BGOUTN	26	42	O	M,MS	**Bus Grant Out:** Direct connect to the selected level among VMEbus BG0OUT* – BG3OUT*.
BBSYN	24	38	O	M,MS	**Bus Busy:** Active-low, open collector direct connect to VMEbus BBSY*.
LBGN	4	5	I/O	M,MS	**Local Bus Grant:** Active-low, open collector. Can be connected to the bus grant input of a DMA controller. Asserted when LBRN is low and the BUSCON has control of the VMEbus. Grounded, or driven low during RESET, to prevent the ASN state being driven onto MASN when the BUSCON is not in control of the VMEbus.
RELSE	10	16	I	M,MS	**Release:** Active-high signal indicating that the onboard logic wants to release control of the VMEbus. In DMA controller applications, the BGACKN output of the DMAC should be connected to (or positive-logic ANDed into) this signal.
DTACKN	19	30	I/O O	M,MS S	**Data Transfer Acknowledge:** Active-low, open collector. Direct connect to VMEbus DTACK*.
BERRN	22	35	I/O O	M,MS S	**Bus Error:** Active-low , open collector. Direct connect to VMEbus BERR*.
LDTACKN	7	11	O, I/O I	M, MS S	**Local DTACK:** Active-low, open collector. Output to onboard master and/or input from onboard slave.
LBERRN	6	9	O, I/O I	M, MS S	**Local Bus Error:** Onboard active-low, open collector. Output to onboard master and/or input from onboard slave.
DSI	3	4	I	All	**Data Strobe:** The high-active or of the onboard data strobes, which may be from the onboard master or VMEbus master.
DSENN	5	8	O	M,MS	**Data Strobe Enable:** Low-active, used to enable the onboard data strobes onto the VMEbus.
R/WN	13	21	I	All	**Read/Write:** Onboard R/W signal from the onboard master or VMEbus master.
DDIR	15	24	O	All	**Data Direction Control:** Direction control for VMEbus data transceivers. A high level indicates the "onboard-to-VMEbus" direction.
DENN	9	15	O	All	**Data Enable:** Low-active enable for VMEbus data transceivers.
RESETN	18	27	I	All	**RESET:** Low-active reset. Clears BUSCON logic.
V_{CC}	8	13	I	All	**Power Supply:** +5V
GND	21	33	I	All	**Ground:** 0V reference.

ADDRESS DECODING

Both the VMEbus and current high-speed processors provide short address-to-strobe setup times, such that with all but the most simple decode schemes, designers must provide for delaying the strobe until decoder outputs have become valid. However, BUSCON operates as a finite-state machine and must synchronize address strobes and other inputs before it can act on them. The BUSCON design allows this synchronization time to be overlapped with address decoding.

In general, most BUSCON inputs do not have critical timing parameters. Exceptions are the three address decode signals. Figure 6 shows a somewhat simplified model of the VMEbus slave selection logic in the SCB68172. The ASN signal is qualified and sampled by flip-flops A and B on each rising edge of CLK. Flip-flop C is set when ASN is high between cycles, and cleared by a falling edge on DTACKN or BERRN.

On the rising edge of CLK after flip-flop B samples ASN low, if C is still set and SLVN is low, flip-flop D is set, indicating slave selection. (In reality, there are more terms in the logic to set D.) Once D is set, it remains set until flip-flop A samples ASN high and a similar circuit (not shown) samples DSI low.

Since SLVN is a direct input to flip-flop D, it must meet a setup time to the clock after ASN is sampled low. Viewed asynchronously, SLVN should be valid slightly less than one clock period after ASN goes low, through shortly after DTACKN goes low.

The onboard logic driven by MASN, VMEN, and ONBD is similar but not as complex. Neither flip-flop C nor a data-strobe-related signal are used, and there are separate flip-flops corresponding to D for each of the VMEN and ONBD signals. VMEN and ONBD should be valid slightly less than one clock period after MASN goes low, through shortly after MASN goes high.

Because ASN and MASN are used directly to clear the corresponding "flip-flop B", their minimum high times are relatively short. However, for consecutive cycles, the inactive time of "flip-flop D" (and signals derived from it) will be at least two clock periods because of the feedback path from "D" to "B".

VMEbus ARBITRATION

BUSCON begins VMEbus arbitration by driving BRN low if MASN, VMEN and ONBD indicate a VMEbus cycle (or if LBRN goes low) and the BUSCON does not have control of the bus.

After driving BRN, BUSCON waits for the BGINN input which is connected to the selected one among the four VMEbus arbitration levels. (During this time it can of course respond to cycles from other VMEbus masters.) When it receives BGINN low while holding BRN low, it drives BBSYN low and thereafter releases BRN. (If it receives

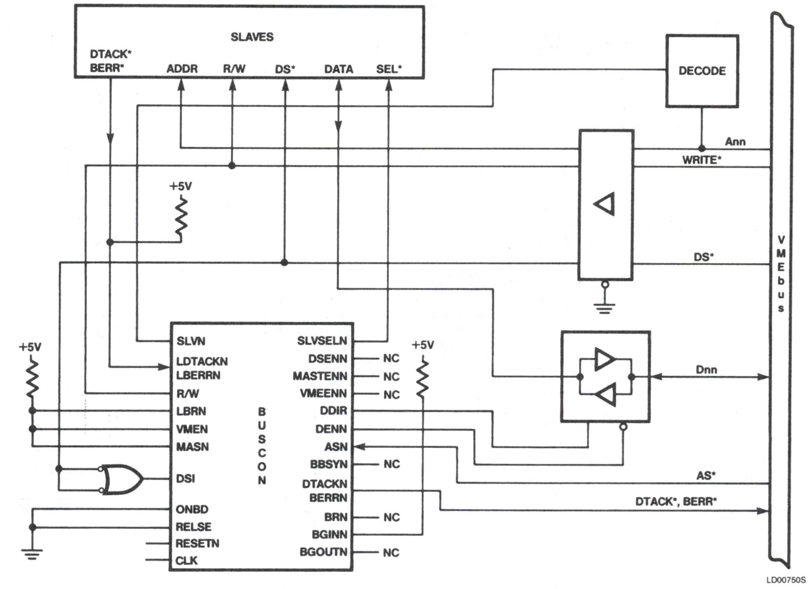

SLAVES
DTACK*
BERR*
ADDR
R/W
DS*
DATA
SEL*
DECODE
Ann
WRITE*
+5V
DS*
V
M
E
b
u
s
+5V
SLVN
LDTACKN
LBERRN
R/W
LBRN
VMEN
MASN
DSI
ONBD
RELSE
RESETN
CLK
B
U
S
C
O
N
SLVSELN
DSENN
MASTENN
VMEENN
DDIR
DENN
ASN
BBSYN
DTACKN
BERRN
BRN
BGINN
BGOUTN
NC
NC
NC
NC
NC
NC
+5V
Dnn
AS*
DTACK*, BERR*
LD00750S

Figure 1. Slave-Only Application

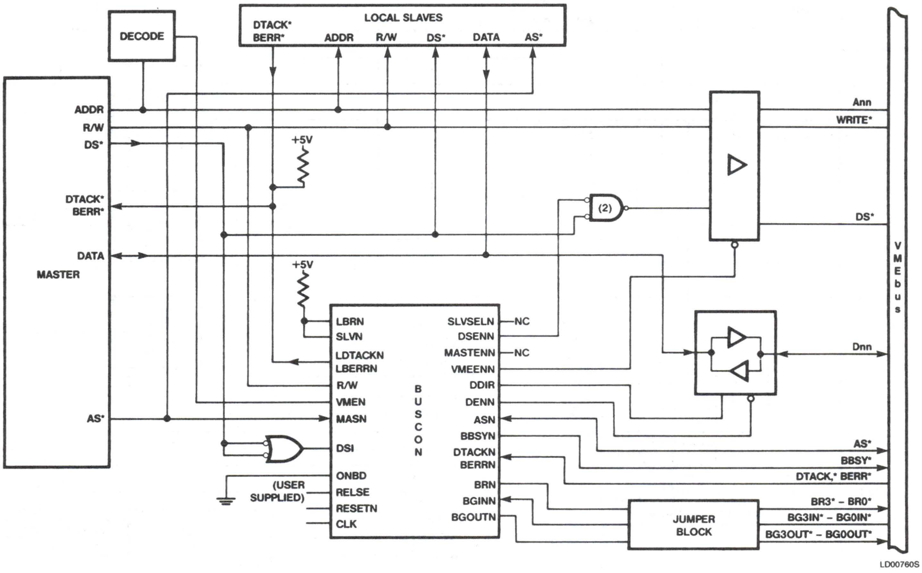

DECODE
LOCAL SLAVES
DTACK*
BERR*
ADDR
R/W
DS*
DATA
AS*
ADDR
R/W
DS*
DTACK*
BERR*
DATA
MASTER
AS*
+5V
+5V
(2)
Ann
WRITE*
DS*
V
M
E
b
u
s
Dnn
LBRN
SLVN
LDTACKN
LBERRN
R/W
VMEN
MASN
DSI
ONBD
RELSE
RESETN
CLK
(USER
SUPPLIED)
B
U
S
C
O
N
SLVSELN
DSENN
MASTENN
VMEENN
DDIR
DENN
ASN
BBSYN
DTACKN
BERRN
BRN
BGINN
BGOUTN
NC
NC
AS*
BBSY*
DTACK,* BERR*
JUMPER
BLOCK
BR3* – BR0*
BG3IN* – BG0IN*
BG3OUT* – BG0OUT*
LD00760S

Figure 2. Master-Only Application

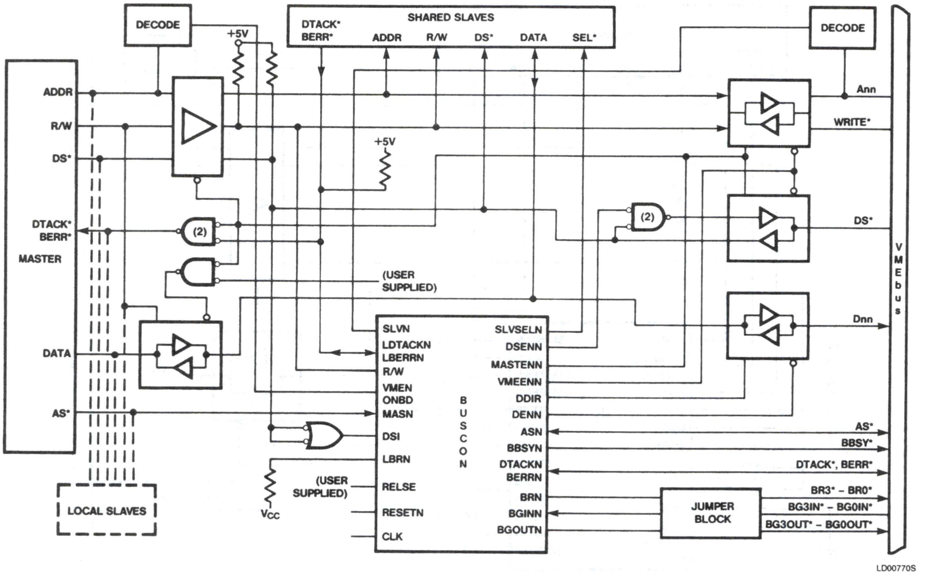

DECODE
+5V
SHARED SLAVES
DTACK*
BERR*
ADDR
R/W
DS*
DATA
SEL*
DECODE
ADDR
R/W
DS*
DTACK*
BERR*
MASTER
DATA
AS*
(2)
(2)
(USER SUPPLIED)
+5V
Ann
WRITE*
DS*
Dnn
VMEbus
SLVN
LDTACKN
LBERRN
R/W
VMEN
ONBD
MASN
DSI
LBRN
RELSE
RESETN
CLK
BUSCON
SLVSELN
DSENN
MASTENN
VMEENN
DDIR
DENN
ASN
BBSYN
DTACKN
BERRN
BRN
BGINN
BGOUTN
AS*
BBSY*
DTACK*, BERR*
BR3* – BR0*
BG3IN* – BG0IN*
BG3OUT* – BG0OUT*
JUMPER BLOCK
LOCAL SLAVES
(USER SUPPLIED)
VCC
LD00770S

Figure 3. Processor-Type Master/Slave Application

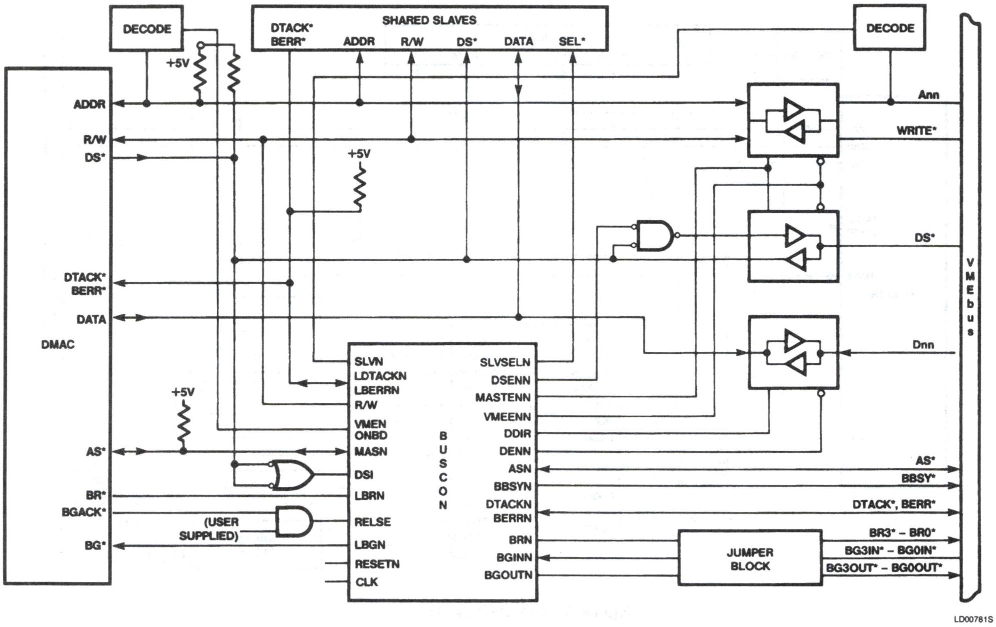

DECODE
+5V
SHARED SLAVES
DTACK*
BERR*
ADDR
R/W
DS*
DATA
SEL*
DECODE
ADDR
R/W
DS*
DTACK*
BERR*
DATA
DMAC
+5V
Ann
WRITE*
DS*
Dnn
VMEbus
+5V
AS*
BR*
BGACK*
BG*
(USER SUPPLIED)
SLVN
LDTACKN
LBERRN
R/W
VMEN
ONBD
MASN
DSI
LBRN
RELSE
LBGN
RESETN
CLK
BUSCON
SLVSELN
DSENN
MASTENN
VMEENN
DDIR
DENN
ASN
BBSYN
DTACKN
BERRN
BRN
BGINN
BGOUTN
AS*
BBSY*
DTACK*, BERR*
BR3* – BR0*
BG3IN* – BG0IN*
BG3OUT* – BG0OUT*
JUMPER BLOCK
LD00781S

Figure 4. DMAC-Type Master/Slave Application

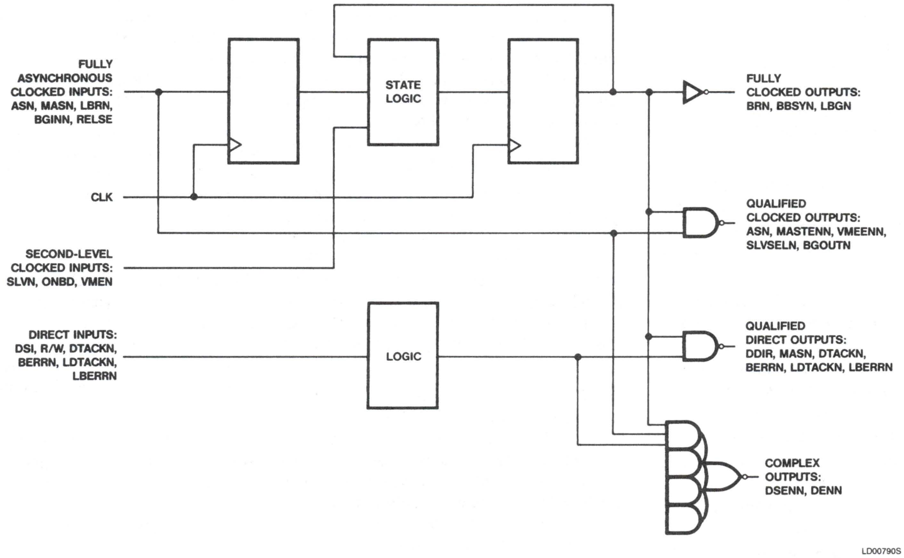

FULLY
ASYNCHRONOUS
CLOCKED INPUTS:
ASN, MASN, LBRN,
BGINN, RELSE
CLK
SECOND-LEVEL
CLOCKED INPUTS:
SLVN, ONBD, VMEN
DIRECT INPUTS:
DSI, R/W, DTACKN,
BERRN, LDTACKN,
LBERRN
STATE
LOGIC
LOGIC
FULLY
CLOCKED OUTPUTS:
BRN, BBSYN, LBGN
QUALIFIED
CLOCKED OUTPUTS:
ASN, MASTENN, VMEENN,
SLVSELN, BGOUTN
QUALIFIED
DIRECT OUTPUTS:
DDIR, MASN, DTACKN,
BERRN, LDTACKN, LBERRN
COMPLEX
OUTPUTS:
DSENN, DENN
LD00790S

Figure 5. Functional Model

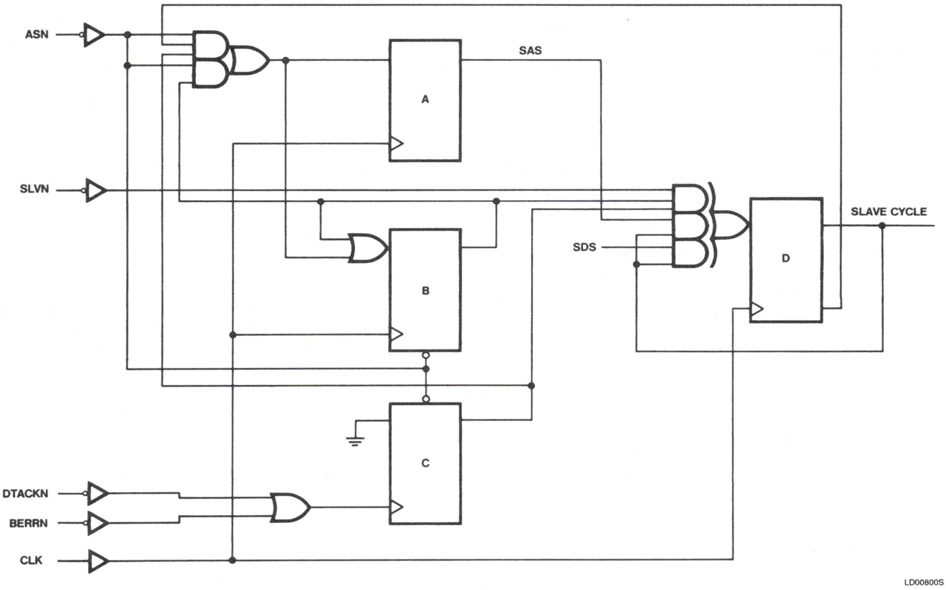

ASN
SLVN
DTACKN
BERRN
CLK
SAS
SDS
A
B
C
D
SLAVE CYCLE
LD00800S

Figure 6. Simplified Model of the SCB68172 Slave Select Logic

BGINN low at any other time, it drives BGOUTN low and continues to do so until BGINN goes high.)

Once BUSCON has driven BBSYN low, it waits for any current VMEbus cycle to complete as evidenced by ASN high. Then it begins to drive ASN (initially high) and drives VMEENN low to enable the address out onto the VMEbus.

If the BRN was initiated by MASN, BUSCON then waits two clock periods for address setup time before driving ASN low. If the bus acquisition was initiated by LBRN, it waits for MASN.

BUSCON will release the BBSYN signal on a rising clock edge at which all of the following conditions are met:

1. It is at least three clock periods after the edge on which BBSYN was asserted, and
2. Any prior master's cycle has completed and VMEENN has been driven low, and
3. The BGINN input was high on the last rising clock edge, and
4. The RELSE input was high on the last rising clock edge, and
5. It is not the clock edge at which BUSCON asserts ASN, and
6. It is not the clock edge at which BUSCON withdraws ASN, and
7. LBRN was high on the last rising clock edge.

If BBSYN is released while BUSCON is not driving ASN low, then VMEENN goes high when BBSYN is released, to release the VMEbus. Otherwise, VMEENN goes high shortly before ASN goes high.

RELSE is provided to allow user determination of the method of VMEbus release. The BGACKN output of a DMA controller can be connected to (or included in) this signal to allow the device to control how long it keeps the bus. The OR of the VMEbus requests can be connected to (or included in) this signal for release on request (ROR) operation. If RELSE is connected to a constant logic high, BUSCON will release the bus as soon as possible; i.e. during the first bus cycle.

VMEbus MASTER OPERATION

When the BUSCON has VMEbus control, VMEbus cycles indicated on MASN and VMEN produce ASN low on the VMEbus. DDIR and DENN control the VMEbus data transceivers. DDIR reflects the R/WN line.

In a write operation, DENN is driven low to drive data onto the VMEbus whenever the BUSCON has control of the VMEbus, R/WN is low, and the previous VMEbus slave has released DTACKN and BERRN to high. (The DTACKN/BERRN requirement does not apply to subsequent cycles among consecutive writes, if R/WN is maintained continuously low.) DSENN is then driven low when DENN has been low for more than a clock period, and DTACKN and BERRN are high, but not before ASN is driven low. DSENN goes high after DSI goes low or MASN goes high, whichever occurs first. DENN goes high after R/WN goes high, or with VMEENN going high, whichever occurs first.

In a read operation (R/WN is high), DENN goes low to drive data in from the VMEbus after ONBD and VMEN have been sampled, DSI is high, and MASTENN is low. DSENN goes low after DSI, DTACKN, and BERRN are all high, but not before ASN goes low. DSENN and DENN go high after DSI goes low or MASN goes high, whichever occurs first.

DTACKN and BERRN are inputs and drive LDTACKN and LBERRN as outputs. LDTACKN and LBERRN are released when the onboard master makes DSI low. If the VMEbus slave continues to hold DTACKN or BERRN low thereafter, DSENN, LDTACKN and LBERRN are inhibited for the next cycle until the response is released.

MASTENN and VMEENN are kept low while the BUSCON has VMEbus control. The MAS-to-AS delay thus provides automatic address-setup time for subsequent VMEbus cycles.

The need to transceive the data strobes in a master/slave application, plus qualify the onboard master's strobes with DSENN for output, can be accomplished in several ways as shown in Figure 7.

MASTER/SLAVE SWITCHING

BUSCON includes arbitration and switching logic between VMEbus slave cycles and onboard master cycles (to a shared onboard slave or the VMEbus). The logic remains in its previous direction until forced to switch by another cycle. This provides minimum overhead for slave-only or master-only operation, and for consecutive cycles from the same master.

If a master cycle to a shared slave occurs, or BGINN arrives when requesting the VMEbus, after a slave cycle with another VMEbus master, VMEENN goes high to disable the address from the VMEbus. On the next clock edge, MASTENN goes low to enable the master's address back out onto the onboard bus.

For a VMEbus master cycle, if the current VMEbus cycle is also over, VMEENN then goes low to enable the address out onto the VMEbus.

For a master cycle to a shared slave, SLVSELN goes low one clock period after MASTENN goes low, or if the master direction is continuing, after ONBD is sampled high. SLVSELN goes high shortly after MASN goes high. DTACKN and BERRN are isolated from LDTACKN and LBERRN. DSENN is kept high. DENN is kept high except in a write cycle when BUSCON has VMEbus control.

If an onboard master cycle and VMEbus slave cycle both arrive for the shared slaves within the same clock period, the previous direction of the master/slave switch is retained.

VMEbus SLAVE OPERATION

If a VMEbus slave cycle occurs after a master cycle, or while BUSCON is requesting the VMEbus, MASTENN goes high, and on the subsequent clock VMEENN goes low to enable the VMEbus address and control signals onto the board.

SLVSELN goes low one clock period after VMEENN goes low, to signal the shared slave(s) that a cycle is occurring. If the slave mode is continuing, SLVSELN goes low after SLVN is sampled low. SLVSELN goes high shortly after ASN goes high.

DDIR reflects R/WN (in the opposite sense from master operation). LDTACKN and LBERRN are inputs and drive DTACKN and BERRN as outputs.

In write operations, DENN is driven low (to enable data in) whenever R/WN is low and LDTACKN and LBERRN are high. When switching between master and slave operation with R/WN low, DENN sequences like VMEENN.

In read operations, DENN is driven low (to enable data out) after SLVN has been sampled low, and while R/WN and DSI are both high.

SLAVE-ONLY USE

This is the simplest application of the BUSCON. However, handling of board-selection logic from a simple VMEbus address decode, plus driving and sequencing of DTACKN and BERRN, can save VMEbus designers cost and board space even in this application.

SLAVE DESIGN

In the MS and S configurations, slaves operate off the data strobes and SLVSELN rather than address and data strobes. It should be noted that SLVSELN will typically go low after the data strobes go low. Data should not be written nor placed on the data lines until SLVSELN goes low.

68000 DUAL-PORTED OPERATION

BUSCON is ideal for use on a VMEbus board containing a 68000 processor. The obvious approach to dual-porting memory and other slaves, on a board with a 68000, is to use the BRN input of the 68000 to suspend processor operation while another master accesses the onboard slave. This works fine except when the processor has already started a cycle for the VMEbus. This latter incident threatens a "deadlock" situation and requires that the processor be "rolled back" off the board's shared bus so that the other master's cycle can occur first. The 68000 has a feature which can be used for this; assertion of both its BERRN and HALTN inputs cause it to suspend operation and retry the cycle when the inputs are released.

The BUSCON does not use these features because there is a flaw in the retry logic. The retry logic does not function during an indivisible RMW sequence (TAS instruction), even in the read cycle. Instead, the assertion of BERRN and HALTN causes an actual bus error exception. It is believed that there is no reliable and general programming solution to the problem of finding the start of the TAS instruction for restart. With 6801x processors, the situation is better because the TAS can be restarted. In any case BUSCON elects to isolate the processor with a few more packages rather than adding complexity to the error-handling software because of dual-ported design.

DMA USE

The BUSCON can be used for VMEbus boards which contain a DMA controller but no processor. Such DMA applications are always master/slave due to the need to program the DMA controller from the VMEbus. There are two operational features of the SCB68172 that are intended for use with DMA controllers. First, the LBRN input can be used to request control of the VMEbus directly, rather than waiting for MASN low and VMEN low as in a processor application. Second, the SCB68172 samples the state of the LBGN pin when RESETN is low. If LBGN is low at the end of RESETN, MASN is used as an input only. If LBGN is high (at the end of the RESETN pulse), the BUSCON thereafter drives the state of VMEbus ASN onto MASN whenever it does not have control of the VMEbus.

For 68000 family DMA controllers, MASN is connected directly to the controller's address strobe pin. The VMEbus ASN-to-MASN feature satisfies the requirement of some DMA controllers that ASN be low on cycles which program them, and also serves to delay the activity of a controller which gets an LBGN response during the last VMEbus cycle by another master.

When LBRN is sampled low, if the BUSCON has retained VMEbus control from previous DMA activity, it continues to retain control for the duration of LBRN being low, and drives LBGN low on the next clock.

Otherwise, it drives VMEbus BRN low on the next clock. When BGINN is sampled low, BBSYN is driven low on the next clock. LBGN is driven low on the same clock as BBSYN if VMEbus ASN is high. If ASN is low, LBGN is

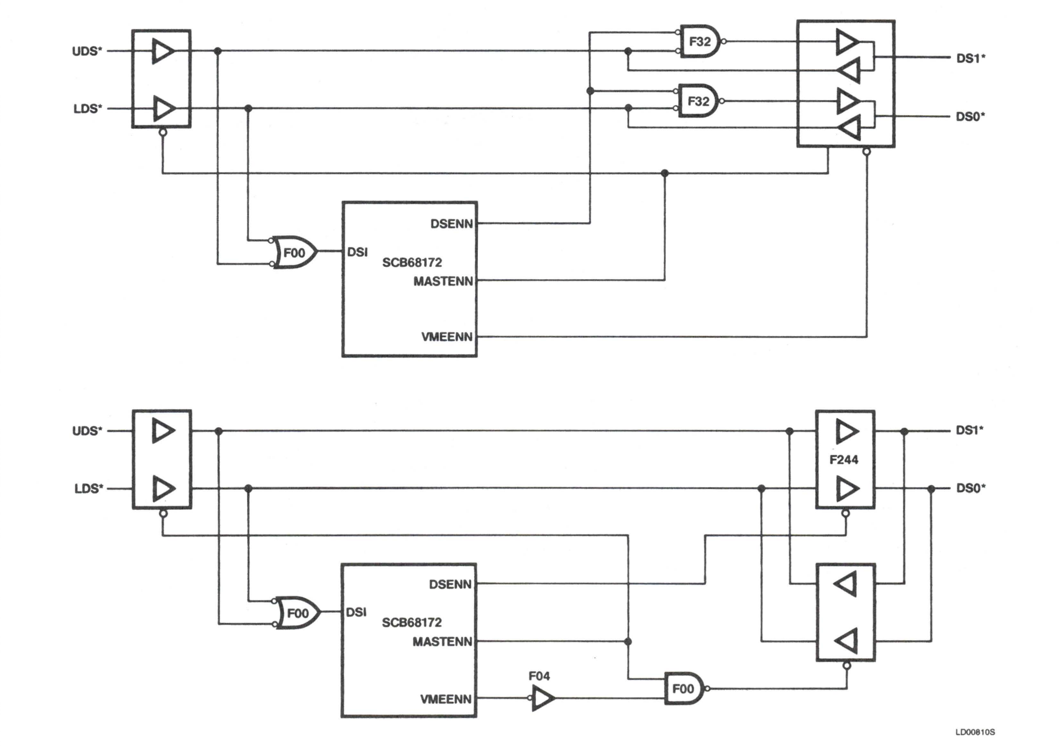


Figure 7. Data Strobe Handling Alternatives in Master/Slave Applications

driven low one clock after BBSYN, except when ASN low and BGINN are both sampled low for the first time in the same clock period and the VMEbus cycle addresses this board – in this last case, LBGN is driven low two clocks after BBSYN. In either of the last two cases, VMEbus ASN low makes MASN low before LBGN goes low, which keeps the DMA controller from starting until the current VMEbus cycle is over.

Note that the local bus request/grant logic and ASN-to-MASN drive features are separate capabilities, either or both of which can be used in applications not involving a DMA controller. However, note also that when the state of the ASN is driven onto MASN, it is done without conditioning by the state of the master-slave switch. This means MASN can go low before MASTENN and VMEENN have been cycled to bring the VMEbus address onto the board. (The low state of SLVSELN indicates that the VMEbus address is valid on the board.)

DMA applications are always considered master/slave due to the need to program the DMA controller. The BUSCON assumes that it must always have VMEbus control before answering an LBRN with LBGN. If this is not desired; i.e., if the DMA controller will sometimes be programmed to do onboard transfers solely and the designer wishes to optimize for this case, then a processor-type interface should be used, and isolation devices and additional onboard logic are required.

INTERRUPT HANDLING

When a processor handles interrupts from onboard sources and from the VMEbus, the design must include logic to decide whether an interrupt acknowledge cycle is an onboard or offboard cycle. This logic is quite different from the address decoding logic used to make this decision on other cycles.

Performance can be maximized if the interrupt logic can provide ONBD and VMEN within the specified time after MASN goes low, or if the signals can be made to meet their specified setup and hold times for CLK. In this case ONBD and VMEN need be selected between the IACK and non-IACK sources. Otherwise (i.e., if the interrupt logic presents these signals slowly and asynchronously), MASN must also be selected between the IACK and non-IACK sources.

MASTER BLOCK TRANSFER

The block transfer feature of the VMEbus allows considerable performance improvement for transferring a block of consecutive memory locations. The BUSCON can be used for block transfer operations in the master role.

Master block transfers are applicable to cache subsystems or block transfer on processor boards, and to DMAC-type designs. The only requirements for master block transfers operation with the BUSCON are that external logic must place a block transfer address modifier (AM) code on the VMEbus, and then hold MASN low across a number of data strobes. (Note that a long block transfer can compromise the operation of other VMEbus masters. One strategy to avoid such problems could be to do a minimum of 4 or 8 transfers without interruption, and then switch to release on request (ROR) operation.)

A sample circuit for master block transfers is shown in Figure 8. Here, a block transfer is triggered whenever the DMAC accesses a certain range of addresses. The SEQ signal could of course be generated in other ways.

SLAVE BLOCK TRANSFERS

VMEbus slaves that are capable of block transfers latch the bus address into a set of counters on the leading edge of ASN, and then increment the address for each data transfer. The SCB68172 can be used on such slave boards in accordance with revision C of the VMEbus specification.

The revision C specification introduces a limitation on block transfers, namely that a master is not allowed to continue a block transfer across a 256-byte boundary. This limitation has a number of advantages, including reducing the number of counter devices needed on slave boards, allowing straightforward use of page or static column modes on dynamic memories, providing periodic windows in a long block transfer in which higher-priority masters can gain bus control, and (effectively) preventing a block transfer from crossing from one slave board to another.

It is this last advantage that is of particular importance for the SCB68172. It means that VMEbus slaves can make a positive selection-decision after ASN goes low, and this decision will remain valid for the duration of the cycle even if it is a block transfer cycle.

In a block transfer which selects an SCB68172-based slave board, SLVSELN remains low through the block, until ASN goes high. The onboard slave logic then uses the data strobes to define each data transfer.

The data strobe and acknowledge signals are handled in a high-speed combinatorial fashion by the SCB68172 in both the master and slave roles. Block transfers are inherently faster on the VMEbus because the address need be passed and decoded only once, and because the slave can look ahead (pipeline) subsequent transfers in a block read cycle. With the SCB68172, this inherent speed advantage is augmented by the advantage of combinatorial over sequential (arbitrated) logic.

3-STATE ENABLE SWITCHING (MASTENN, VMEENN, DENN)

As a result of speed optimization of master/slave switching, some parts used in PMS applications may exhibit short high-going transients on MASTENN, VMEENN, and/or DENN, if requests for access to the shared slave(s) arrive closely in time from both the onboard master and the VMEbus master. These transients should pose no problem as long as the signals are used as intended (i.e., as 3-State enables). The following points apply:

1. A transient will occur only when SLVSELN has been high for at least one clock period, and at least one clock period before a subsequent low on SLVSELN.
2. A transient will occur only if the current master/slave direction is maintained.
3. Low-going transients (which could cause tristate conflicts) do not occur.
4. Commonly such transients will be eliminated by external capacitance, and/or rejected by receivers on other parts. In any case the logic levels on signals controlled by these enable signals should not be affected.
5. Edge-sensitive use of these signals is inadvisable in a PMS application.

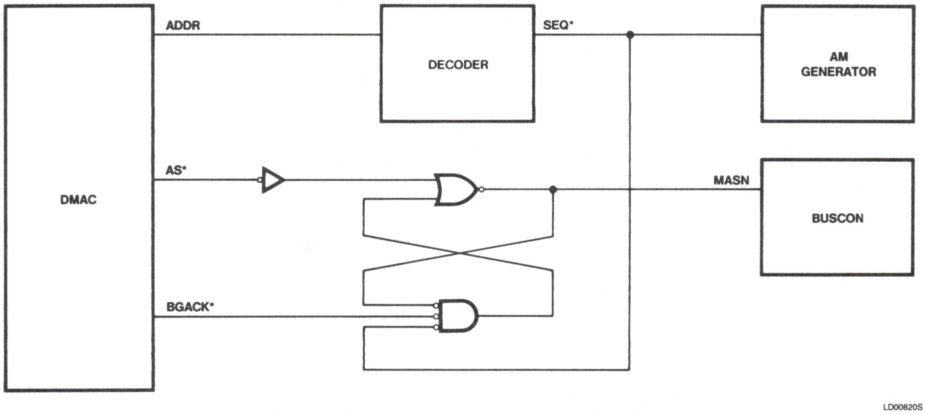
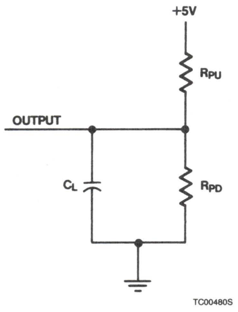

DMAC
ADDR
DECODER
SEQ*
AM GENERATOR
AS*
MASN
BUSCON
BGACK*
LD00820S

Figure 8. Master Sequential Mode With DMAC

TEST CONDITIONS

Unless otherwise noted, the following timing parameters are based on loading as follows:


+5V
R_{PU}
OUTPUT
C_L
R_{PD}
TC00480S

SIGNALS	R_{PU}	R_{PD}	C_L
ASN, BRN, BBSYN, DTACKN, BERRN LDTACKN, LBERRN, LBGN, BGOUTN, VMEENN, SLVSELN, DSENN, DENN,	150	235	300
DDIR	2K	N/A	15
MASTENN	280	N/A	45
MASN	180	1K	45

VMEbus Controller (BUSCON) SCB68172

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
V_S	Supply voltage range	-0.5 to +7	V
V_{IN}	Input voltage range	-0.5 to +5.5	V
T_A	Operating temperature range[2]	0 to +70	°C
T_{STG}	Storage temperature range	-65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to $+70$ °C[3, 4]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
DSI, ONBD, SLVN, VMEN, LBRN, RELSE, RESETN					
I_{IL}	Input low current	$V_{IN} = 0.4V$		-400	µA
I_{IH}	Input high current	$V_{IN} = 2.7V$		20	µA
V_{IL}	Input low voltage			0.8	V
V_{IH}	Input high voltage		2		V
ASN, MASN, BGINN, DTACKN, BERRN, LDTACKN, LBERRN					
I_{IL}	Input low current	$V_{IN} = 0.4V$		-400	µA
I_{IH}	Input high current	$V_{IN} = 2.7V$		20	µA
V_{TH+}	High-going threshold voltage		1	1.65	V
V_{TH-}	Low-going threshold voltage		0.8	1.15	V
$V_{TH+} - V_{TH-}$	Hysteresis		0.2		V
R/WN, CLK					
I_{IL}	Input low current	$V_{IN} = 0.4V$		-800	µA
I_{IH}	Input high current	$V_{IN} = 2.7V$		40	µA
V_{IL}	Input low voltage			0.8	V
V_{IH}	Input high voltage		2		V
BGOUTN, VMEENN, SLVSELN, DSENN, DENN, DDIR (low current totem pole)					
V_{OL}	Output low voltage	$I_{OL} = 8mA$, $V_{CC} = 4.75V$		0.5	V
V_{OH}	Output high voltage	$I_{OH} = -0.4mA$, $V_{CC} = 4.75V$	2.7		V
I_{OS}	Short-circuit output current	$V_{OUT} = 0V$	-15	-100	mA
MASTENN (high current totem pole)					
V_{OL}	Output low voltage	$I_{OL} = 24mA$, $V_{CC} = 4.75V$		0.5	V
V_{OH}	Output high voltage	$I_{OH} = -2.6mA$, $V_{CC} = 4.75V$	2.4		V
V_{OH}	Output high voltage	$I_{OH} = -1mA$, $V_{CC} = 4.75V$	2.7		V
I_{OS}	Short-circuit output current	$V_{OUT} = 0V$	-40	-100	mA
MASN (low current tristate)					
V_{OL}	Output low voltage	$I_{OL} = 8mA$, $V_{CC} = 4.75V$		0.5	V
V_{OH}	Output high voltage	$I_{OH} = -0.4mA$, $V_{CC} = 4.75V$	2.7		V
I_{OS}	Short-circuit output current	$V_{OUT} = 0V$	-15	-100	mA
I_{OZL}	3-State-off leakage current, low level	$V = 0.4V$		-21	µA
I_{OZH}	3-State-off leakage current, high level	$V = 2.7V$		20	µA
ASN (high current tristate)					
V_{OL}	Output low voltage	$I_{OL} = 64mA$, $V_{CC} = 4.75V$		0.5	V
V_{OH}	Output high voltage	$I_{OH} = -7.8mA$, $V_{CC} = 4.75V$	2.4		V
V_{OH}	Output high voltage	$I_{OH} = -3mA$, $V_{CC} = 4.75V$	2.7		V
I_{OS}	Short-circuit output current	$V_{OUT} = 0V$	-120	-300	mA
I_{OZL}	3-State-off leakage current, low level	$V = 0.4V$		-60	µA
I_{OZH}	3-State-off leakage current, high level	$V = 2.7V$		60	µA

DC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
LDTACKN, LBERRN, LBGN (low current open collector)					
V_{OL}	Output low voltage	I_{OL} = 8mA, V_{CC} = 4.75V		0.5	V
I_{OH}	Output leakage current	V = 5.5V		100	μA
DTACKN, BERRN, BRN, BBSYN (high current open collector)					
V_{OL}	Output low voltage	I_{OL} = 40mA, V_{CC} = min		0.4	V
V_{OL}	Output low voltage	I_{OL} = 70mA, V_{CC} = min		0.5	V
I_{OH}	Output leakage current	V = 2.7V		60	μA
I_{OH}	Output leakage current	V = 5.5V		250	μA
I_{CC}	V_{CC} Supply current	V_{CC} = Max		180	mA

NOTE:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other conditions other than those indicated in the Electrical Characteristics section of this data sheet is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. Parameters are valid over specified temperature range.
4. All voltage measurements are referenced to ground (V_{SS}). For testing, all signals swing between 0.4V and 2.4V with a transition time of 10ns maximum. All time measurements are referenced at input voltages of 0.8V and 2V as appropriat e.

AC ELECTRICAL CHARACTERISTICS V_{CC} = 5V ±5%, T_A = 0°C to +70°C[3, 4]

NO.	FIGURE	CHARACTERISTIC	TENTATIVE LIMITS		UNIT	NOTES
			Min	Max		
Clock and general parameters						
*1	9 – 18, 20	CLK cycle time (clk)	40		ns	
*2	9 – 18, 20	CLK low time	15		ns	
*3	9 – 18, 20	CLK high time	15		ns	
Asynchronous input setup time to CLK high						
4	9, 10, 11, 12, 13, 18, 20	ASN, MASN low	15		ns	5
5	9, 10, 11, 13	ASN, MASN high	14		ns	5
*6	9, 10, 11, 12, 13, 20	SLVN, VMEN low	25		ns	6
*7	11, 17, 18	SLVN, VMEN high	20		ns	6
*8	9, 10, 13	ONBD low	25		ns	6
9	18	ONBD high	25		ns	6
10	13, 16, 20	LBRN, RELSE, BGINN low	9		ns	5
11	14, 15	LBRN, RELSE, BGINN high	11		ns	5
12	11, 12, 13	DSI low (end of slave cycle)	16		ns	5
Asynchronous input hold time from CLK high						
13		ASN, MASN, DSI	0		ns	7
14		ONBD, VMEN	0		ns	8
15		LBRN, RELSE, BGINN	2		ns	7
Propagation, CLK high to:						
16	16	BGOUTN low	12	27	ns	
17	20	LBGN low	12	30	ns	
18		LBGN high	16	41	ns	
19	13, 16, 20	BBSYN, BRN low	17	37	ns	
20	13, 20	BBSYN, BRN high (C_L = 50)	20	47	ns	
		(C_L = 300)	40	68	ns	
21	9, 10, 13	ASN low	15	37	ns	
22	9, 10	ASN high	13	31	ns	
23	11, 12, 13, 18, 20	SLVSELN, VMEENN low	14	35	ns	
24	13, 18, 20	MASTENN low	15	38	ns	
Miscellaneous						
*25		RESETN width low	6clk		ns	9
26	16	BGINN low to BGOUTN low	clk+10	2clk+35	ns	
27	16	BGINN high to BGOUTN high	3	10	ns	
*28	9, 11	R/WN high to DSI high (start of read cycle)	10		ns	
*29	9, 12	DSI low to RWN low (end of read cycle)	10		ns	

NOTE:

* These AC Electrical Characteristics have been tested and characterized, and therefore are considered limits, not tentative limits.

VMEbus Controller (BUSCON) SCB68172

AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	FIGURE	CHARACTERISTIC	TENTATIVE LIMITS		UNIT	NOTES
			Min	Max		
Address decoding						
30	11, 12, 13, 20	ASN low to SLVN valid		clk–5	ns	
31	9, 10, 13, 18	MASN low to VMEN valid		clk–5	ns	
32	9, 10, 13, 18	MASN low to ONBD valid		clk–7	ns	
33	17	SLVN high after DTACKN low	14		ns	
34	9, 10	VMEN, ONBD valid after MASN high	6		ns	
35	9, 10	MASN high	15		ns	10
36	9, 10, 11	DSI low	20		ns	
37	11	ASN high	20		ns	10
VMEbus acquisition						
38	13	MASN low to BRN low	clk+15	2clk+45	ns	11
*38A	13, 20	ASN low to BGINN low (early release by other master)	10		ns	
39	13, 20	BGINN low to BBSYN low	clk+15	2clk+45	ns	
40	13, 20	BBSYN low to BRN high	0	50	ns	
41	13	BGINN low to VMEENN low, DENN low (write)	clk+12	2clk+40	ns	12
42	13	ASN high to VMEENN low, DENN low (write)	11	31	ns	12,13
43	13	VMEENN low to ASN low	2clk–10	2clk+15	ns	14
*43A	13, 16, 20	BBSYN or BGOUTN low to BGINN high	0		ns	
VMEbus master cycles						
44	9, 10	ASN high (successive VMEbus master cycles)	2clk–15		ns	14
45	9, 10	MASN low to ASN low (subsequent cycle retaining VMEbus control)	clk+13	2clk+45	ns	14
46	9, 10	ASN low to DSENN low	–4	5	ns	17, 18
47	10, 13	R/WN low to DDIR high	6	15	ns	
48	10, 13	DDIR high to DENN low (write)	2	7	ns	15
49	10	DTACKN and BERRN high to DENN low (write, 1st bus cycle or preceded by read)	7	21	ns	15
50	10	DENN low to DSENN low (write)	clk+10	2clk+40	ns	17
51	9, 11, 13	R/WN high to DDIR low	6	16	ns	
52	9	DDIR low to DENN low (read)	5		ns	16
53	9	DSI high to DENN low (read)	8	18	ns	16
54	9	DSI high to DSENN low (read)	8	18	ns	18
55	9, 10	DTACKN and BERRN high to DSENN low	6	19	ns	17, 18
56	9, 10	DTACKN or BERRN low to LDTACKN or LBERRN low	6	17	ns	
*57	9, 10	LDTACKN or LBERRN low to DSI low or MASN high	0		ns	
58	9, 10	DSI low to DSENN high	7	16	ns	19
59	9, 10	MASN high to DSENN high	13	28	ns	
60	10	R/WN high to DSENN high (after a write)	5	14	ns	20
61	9, 10	MASN high to ASN high (unless early release)	clk+11	2clk+38	ns	
62	9	MASN high to DENN high (read)	13	28	ns	21
63	9	DSI low to DENN high (read)	7	16	ns	21
64	10	R/WN high to DENN high (write)	5	14	ns	22
65	9, 10	DSENN high to LDTACKN and LBERRN high	7	24	ns	23
66	9, 10	DTACKN and BERRN high to LDTACKN and LBERRN high	7	23	ns	23
VMEbus release						
*67	14	BBSYN low	2clk		ns	
68	14, 15	BGINN high to BBSYN high	clk+18	2clk+70	ns	24
69	14, 15	RELSE high to BBSYN high	clk+20	2clk+72	ns	24
70	14	ASN low to BBSYN high (early release)	clk–25		ns	
71	14	MASN high to VMEENN high (early release)	8	20	ns	
72	14	MASN high to DENN high (early release, write)	8	20	ns	
73	14	DENN (write) and VMEENN high to ASN high (early release)	5	15	ns	
74	14	ASN high to ASN released (early release)	5	20	ns	
75	15	ASN high to BBSYN high (intercycle release)	clk–10		ns	
76	15	DENN (write) and VMEENN high to BBSYN high (intercycle release)	5	30	ns	
*77	14, 15	BBSYN high to RELSE low	0		ns	

NOTE:

* These AC Electrical Characteristics have been tested and characterized, and therefore are considered limits, not tentative limits.

AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	FIGURE	CHARACTERISTIC	TENTATIVE LIMITS		UNIT	NOTES
			Min	Max		
Master to slave switching						
78	11, 13, 20	(External) ASN low to MASTENN high	10	clk+40	ns	25
79	11, 13, 20	SLVN Low to MASTENN high	7	17	ns	25
80	11	SLVSELN high to MASTENN high	7		ns	25
81	11, 20	MASTENN high to VMEENN low	14	clk+36	ns	
81A	11, 13	VMEENN low to DDIR change	−5	+5	ns	
82	11, 13, 20	VMEENN low to SLVSELN low	clk−11	clk-5	ns	26
VMEbus slave cycles						
83	12	SLVSELN high (successive slave cycles)	2clk−5		ns	26
84	12	ASN low to SLVSELN low (already in slave state)	clk+12	2clk+40	ns	26
85	12, 13	R/WN low to DDIR low	6	14	ns	
86	12, 13	DDIR low to DENN low (write)	5	12	ns	27
87	12	LDTACKN and LBERRN high to DENN low (write)	7	20	ns	27
88	12, 13	R/WN high to DDIR high	5	14	ns	
89	11	DDIR high to DENN low (read)	2	7	ns	28
90	11	ASN low to DENN low (read)	clk+20	2clk+47	ns	28
91	11	DSI high to DENN low (read)	6	16	ns	28
*92	11, 12, 13, 18	SLVSELN low and DSI high to LDTACKN or LBERRN low	0		ns	29
93	11, 12, 13, 18	LDTACKN or LBERRN low to DTACKN or BERRN low	10	20	ns	
94	12, 13	LDTACKN or LBERRN low to DENN high (write)	9	22	ns	
*94A	11, 12, 13, 18	DTACKN or BERRN low to DSI low or ASN high	0		ns	
95	11	DSI low to DENN high (read)	7	16	ns	
96	11, 12, 18, 20	ASN high to SLVSELN high	13	28	ns	
97	11, 12, 13	DSI low to DTACKN and BERRN high				
		(C_L = 50)	17	37	ns	
		(C_L = 300)	37	60	ns	
*98	11, 12, 13, 20	DSI low to LDTACKN and LBERRN high	0	35	ns	30, 32
*99	11, 12	LDTACKN and LBERRN high to (next) DSI high	0		ns	30
Slave to master switching						
100	18	MASN low to VMEENN, DENN (VMEbus slave write) high	18	clk+47	ns	31
101	18	ONBD high to VMEENN, DENN (VMEbus slave write) high	8	24	ns	31
102		VMEN low to VMEENN, DENN (VMEbus slave write) high	18	21	ns	31
*103	13, 18, 20	SLVSELN high to VMEENN, DENN (VMEbus slave write) high	0		ns	31
104	13, 20	DSI low (selected) to VMEENN, DENN (VMEbus slave write) high	14	clk+40	ns	31
105	13, 18, 20	VMEENN high to MASTENN low	24	clk+20	ns	
107	13, 20	MASTENN low to VMEENN low, DENN low (write, if next cycle on VMEbus)	3		ns	12,15
108	18	MASTENN low to SLVSELN low (if next cycle on board)	clk−13	clk	ns	33
Onboard cycles						
109	18	SLVSELN high (successive onboard cycles)	3clk+4		ns	33
110	18	MASN low to SLVSELN low (MASTENN already low)	clk+17	2clk+42	ns	33
111	18	MASN high to SLVSELN high	12	26	ns	

NOTE:

* These AC Electrical Characteristics have been tested and characterized, and therefore are considered limits, not tentative limits.

VMEbus Controller (BUSCON) SCB68172

AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	FIGURE	CHARACTERISTIC	TENTATIVE LIMITS		UNIT	NOTES
			Min	Max		
DMAC-Type operation						
112	19	BBSYN high to MASN active (C_L = 50)	0	5	ns	34
		(C_L = 300)	−25	−15	ns	34
113	19	ASN high to MASN active	13	40	ns	34
114	19	ASN low to MASN low	10	25	ns	
115	19	ASN high to MASN high	6	16	ns	
116	19	ASN high to MASN released	12	32	ns	35
117	19	LBGN low to MASN released	5	12	ns	35
118	20	LBRN low to BRN low (if BBSYN released)	clk+15	2clk+45	ns	
119		LBRN low to LBGN low	clk+10	2clk+37	ns	36
120	20	BGINN low to LBGN low (ASN high)	clk+15	2clk+45	ns	
		(ASN low)	2clk+15	3clk+45	ns	
121	20	MASN low (output) to LBGN low	2clk+12		ns	
*122	20	LBGN low to LBRN high	0		ns	
123	20	LBRN high to LBGN high	clk+14	2clk+48	ns	37
124	14, 15	LBRN high to BBSYN high (C_L = 50)	clk+18	2clk+55	ns	24
		(C_L = 300)	clk+35	2clk+75	ns	24
125	20	ASN high to LBGN high (selected)	2clk+18	3clk+50	ns	37
126	20	DSI low to LBGN high (selected)	2clk+20	3clk+52	ns	37

NOTE:

* These AC Electrical Characteristics have been tested and characterized, and therefore are considered limits, not tentative limits.

NOTES:

5. These setup times guarantee recognition at a rising edge of CLK, but the device will operate correctly if they are not met. If the asynchronous input is changed between the setup and hold times, the new state of the input may be recognized at this clock or the following clock.
6. These setup times are required on the rising edge of CLK following the one on which ASN or MASN is first recognized low. If parameters 30, 31, and 32 are met, these parameters are automatically guaranteed.
7. These hold times guarantee (continued) recognition of the signal state at a rising edge of CLK, but the device will operate correctly if they are not met.
8. These hold times are required on the rising edge of CLK preceding the one on which MASN is first recognized high. Parameter 34 provides a more straightforward requirement which guarantees these times.
9. This parameter applies after V_{CC} and the clock signal are both within the specified limits.
10. These minimum times are to guarantee recognition. Operation will be limited by 44, 83, and 109 if the strobe is high for less than 2clk.
11. BRN is driven low, and this acquisition sequence applies, only if BBSYN is high.
12. VMEENN is driven low only when 41, 42, and 107 have been met.
13. Applies to ASN of VMEbus cycle which does not select this board as a slave.
14. ASN goes low when 43, 44, and 45 are met. 44 is not applicable on the first cycle after acquiring the VMEbus.
15. In a write operation, DENN goes low when 41, 42, 48, 49, and 107 are met. 49 does not apply for subsequent cycles in a series of writes if R/WN is held low throughout.
16. In a read operation, DENN goes low when 52 and 53 are met.
17. In a write operation, DSENN goes low when 46, 50, and 55 are met. 55 is significant only for subsequent cycles in a series of writes with R/WN held low throughout.
18. In a read operation, DSENN goes low when 46, 54, and 55 are met.
19. DSENN goes high when either 58 or 59 is met.
20. Applies only if R/WN remains low after a write cycle, so that DSENN goes low again.
21. In a read operation, DENN goes high when either 63 or 64 is met.
22. In a write operation, DENN goes high when either 64 or 71 is met.
23. LDTACKN and LBERRN go high when either 65 or 66 is met.
24. BBSYN is always released in response to BGINN, RELSE, and LBRN all high. However, if this condition is detected during a clock period in which the decision to change ASN is made, the release of BBSYN is delayed one clock period so that 70 or 75 is met.
25. MASTENN goes high when 78, 79, and 80 are all met.
26. SLVSELN goes low when 82, 83, and 84 (as applicable) are met.
27. In a write operation, DENN goes low when 86 and 87 are met.
28. In a read operation, DENN goes low when 89, 90, and 91 are met.
29. The onboard slave(s) should wait for both SLVSELN and DSI before driving a response.
30. Since BUSCON itself terminates DTACKN and BERRN when DSI goes low, the onboard slaves must meet this requirement to assure that a "lingering" LDTACKN or LBERRN is not presented as DTACKN or BERRN when the next DSI occurs. 99 is the real requirement – 98 max is derived from it, plus the 40ns minimum high time of VMEbus data strobes and an allowance for receiver skew.
31. VMEENN goes high only when 100, (101 or 102), 103, and 104 are met. 104 applies only if a VMEbus slave cycle (with this board) is ending.
32. The onboard slave(s) must meet this requirement so that the local response is not inadvertantly presented to the onboard master.
33. SLVSELN goes low when 108, 109, and 110 (as applicable) are met.
34. MASN is driven out of Hi-Z state only when 112 and 113 are met.
35. MASN is released to Hi-Z state only when 116 and 117 are met.
36. The max Figure applies only if BUSCON has kept VMEbus control (i.e., if BBSYN is low).
37. LBGN goes high only when 123, 125, and 126 are met, but 125 and 126 apply only if a VMEbus slave cycle (with this board) is in progress.

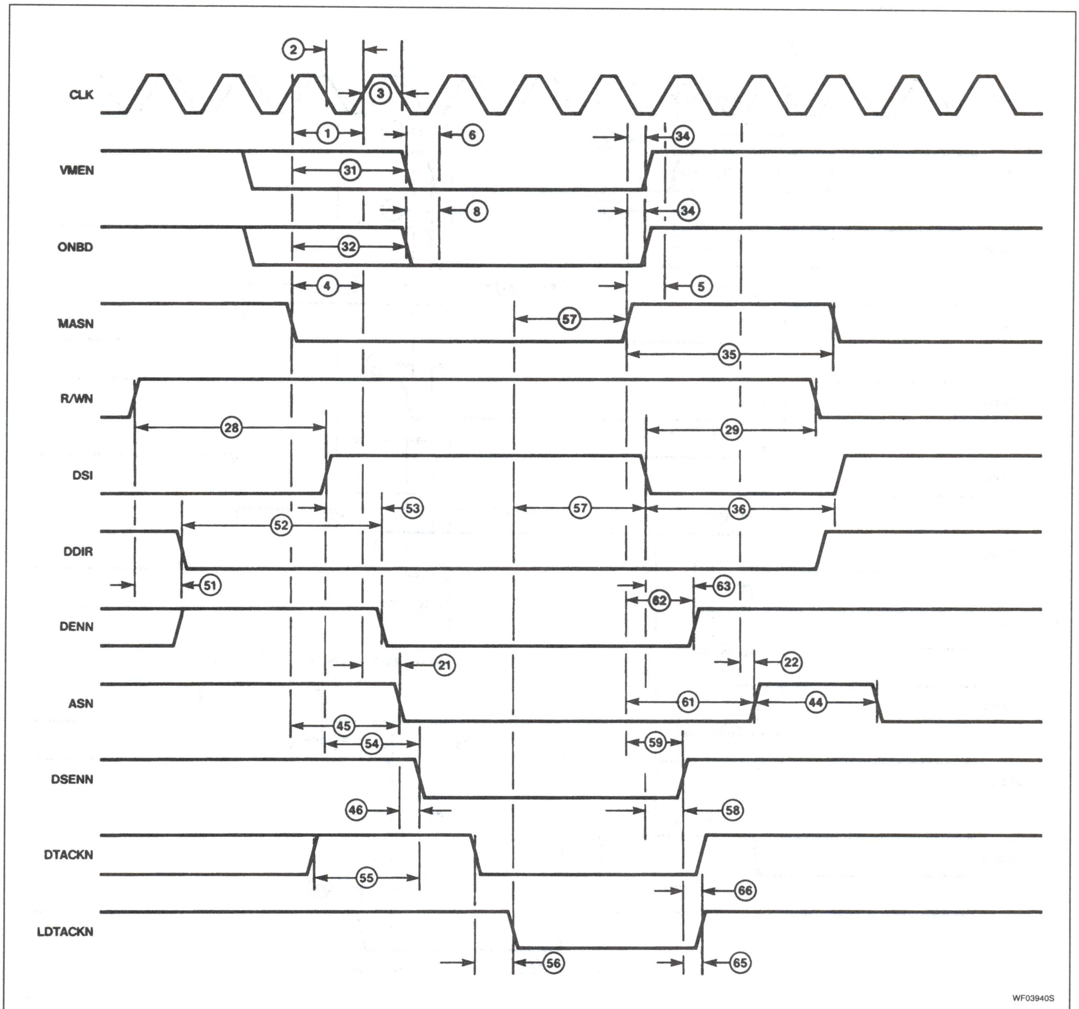

CLK
VMEN
ONBD
MASN
R/WN
DSI
DDIR
DENN
ASN
DSENN
DTACKN
LDTACKN
1
2
3
4
5
6
8
21
22
28
29
31
32
34
34
35
36
44
45
46
51
52
53
54
55
56
57
57
58
59
61
62
63
65
66
WF03940S

Figure 9. Read Cycle From Onboard Master to VMEbus Slave (Already in Control of VMEbus)

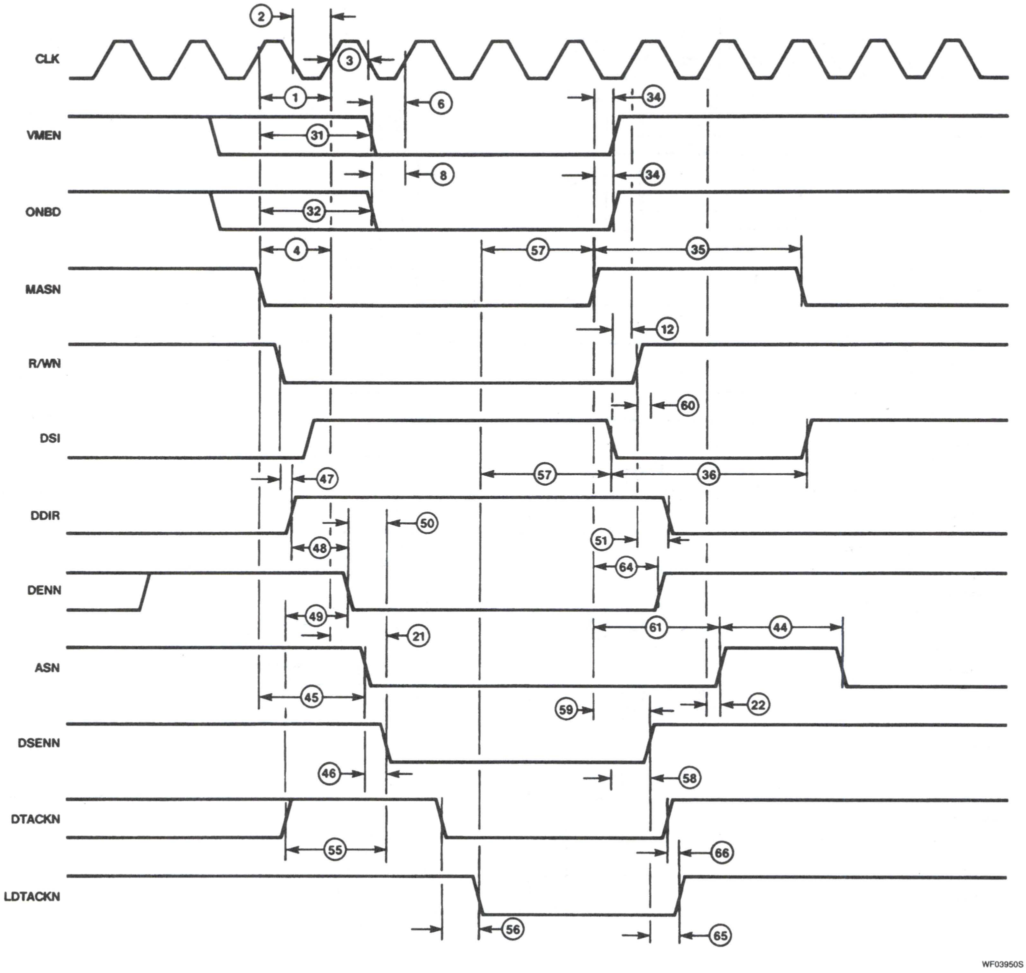

CLK
VMEN
ONBD
MASN
R/WN
DSI
DDIR
DENN
ASN
DSENN
DTACKN
LDTACKN
2
3
1
6
34
31
8
34
32
4
57
35
12
60
57
36
47
50
48
51
64
49
21
61
44
45
59
22
46
58
55
66
56
65
WF03950S

Figure 10. Write Cycle From Onboard Master to VMEbus Slave (Already in Control of VMEbus)

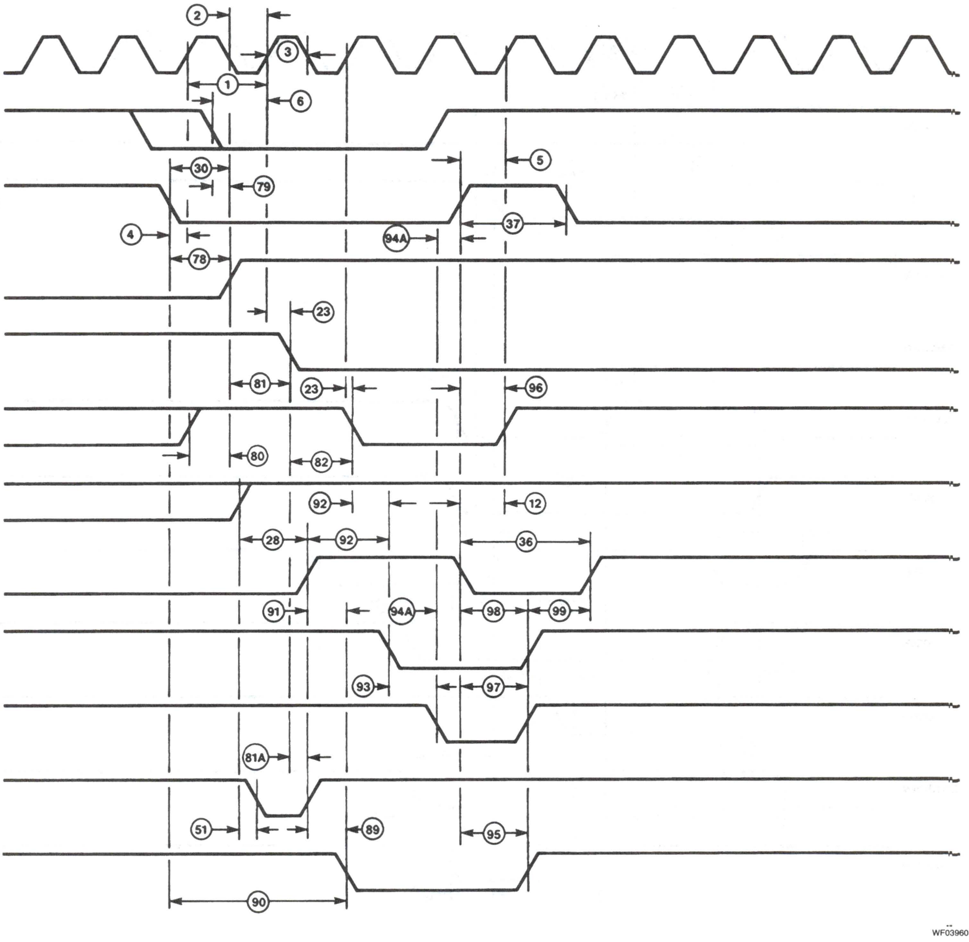

CLK
SLVN
ASN
MASTENN
VMEENN
SLVSELN
R/WN
DSI
LDTACKN
DTACKN
DDIR
DENN
2
3
1
6
5
30
79
37
4
94A
78
23
81
23
96
80
82
92
12
28
92
36
91
94A
98
99
93
97
81A
51
89
95
90
WF03960S

Figure 11. Slave Read Cycle (Starting in Master Mode)

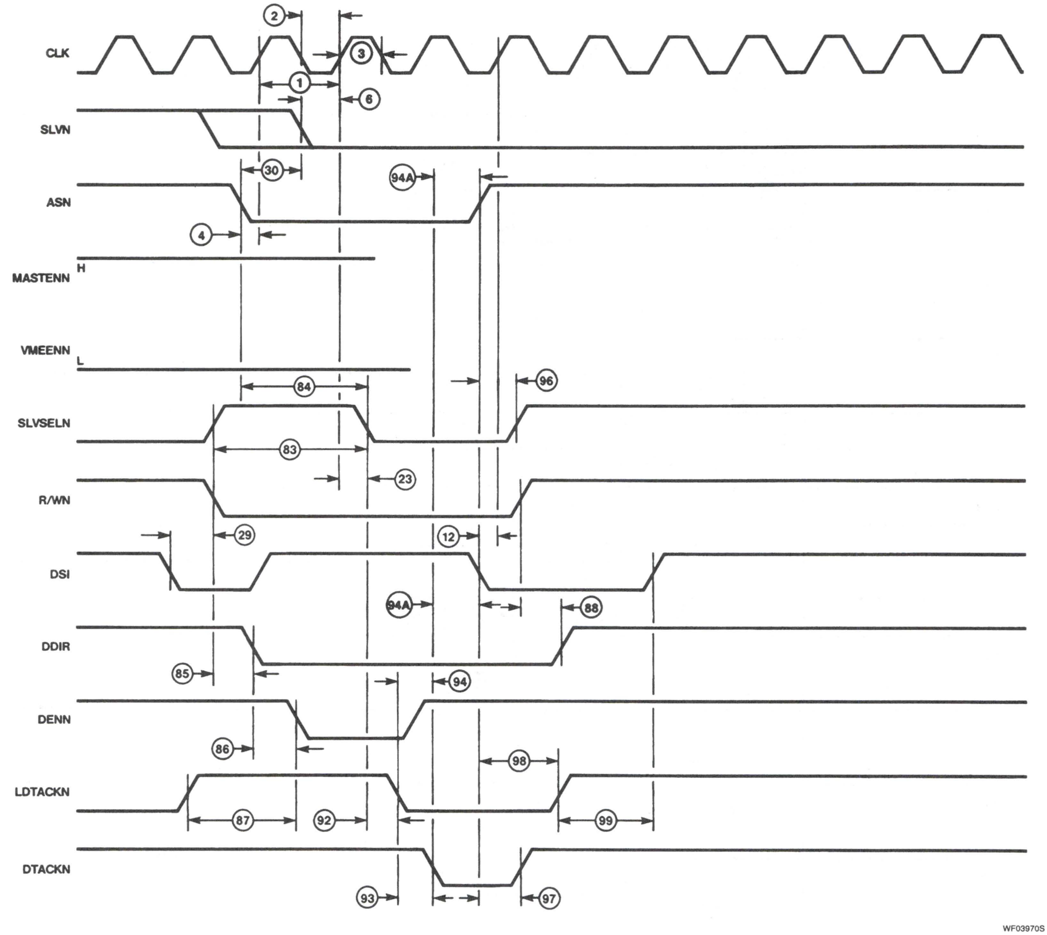

CLK
SLVN
ASN
MASTENN
H
VMEENN
L
SLVSELN
R/WN
DSI
DDIR
DENN
LDTACKN
DTACKN
②
③
①
⑥
㉚
94A
④
84
96
83
㉓
㉙
⑫
88
94A
85
94
86
98
87
92
99
93
97
WF03970S

Figure 12. Slave Write Cycle (Starting in Slave Mode)

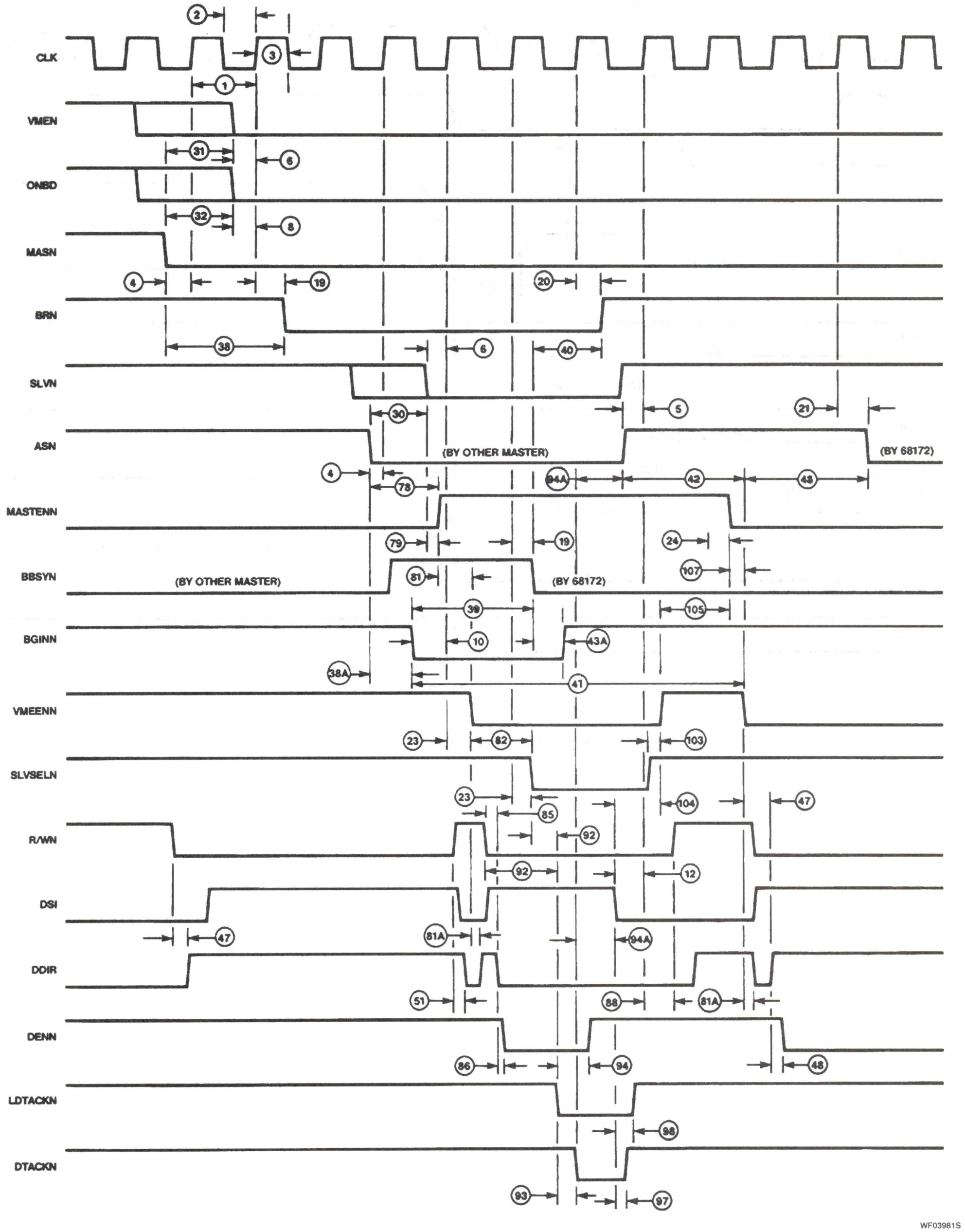

CLK
VMEN
ONBD
MASN
BRN
SLVN
ASN
MASTENN
BBSYN
BGINN
VMEENN
SLVSELN
R/WN
DSI
DDIR
DENN
LDTACKN
DTACKN
(BY OTHER MASTER)
(BY 68172)
(BY OTHER MASTER)
(BY 68172)
①
②
③
④
⑤
⑥
⑧
⑩
⑫
⑲
⑳
㉑
㉓
㉔
㉚
㉛
㉜
㊳
38A
㊴
㊵
㊶
㊷
㊸
43A
㊼
㊽
(51)
(78)
(79)
(81)
81A
(82)
(85)
(86)
(88)
(92)
(93)
(94)
94A
(97)
(98)
(103)
(104)
(105)
(107)
WF03981S

Figure 13. VMEbus Acquisition, Including Slave Write Cycle

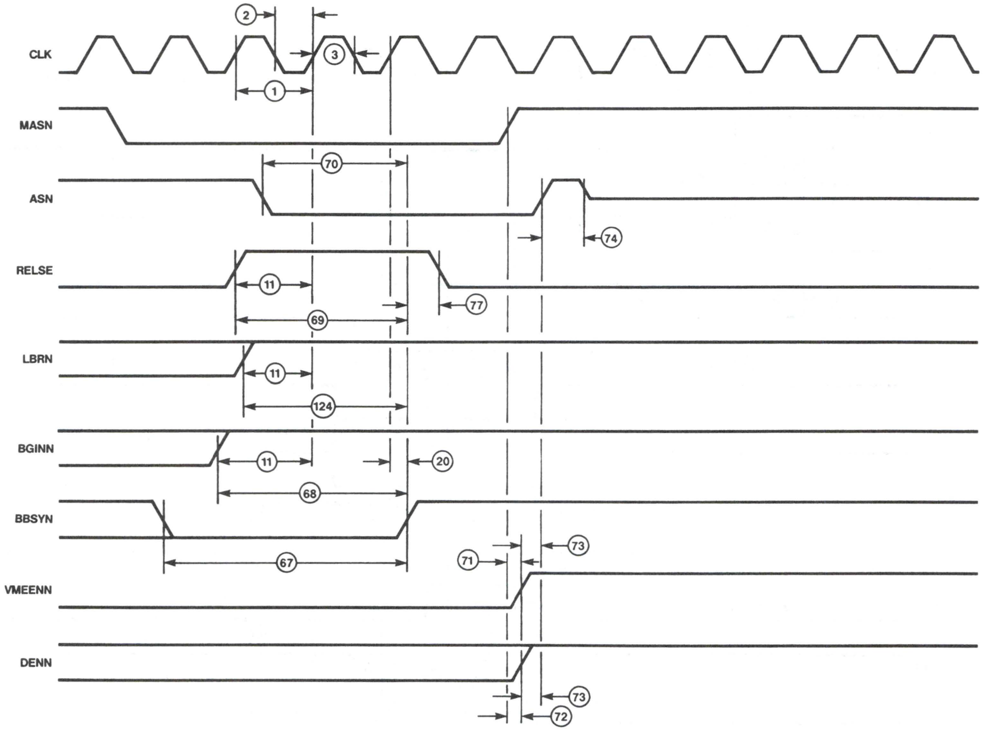

CLK
MASN
ASN
RELSE
LBRN
BGINN
BBSYN
VMEENN
DENN
②
③
①
70
74
11
77
69
11
124
11
20
68
67
71
73
73
72
WF03990S

Figure 14. "Early" VMEbus Release

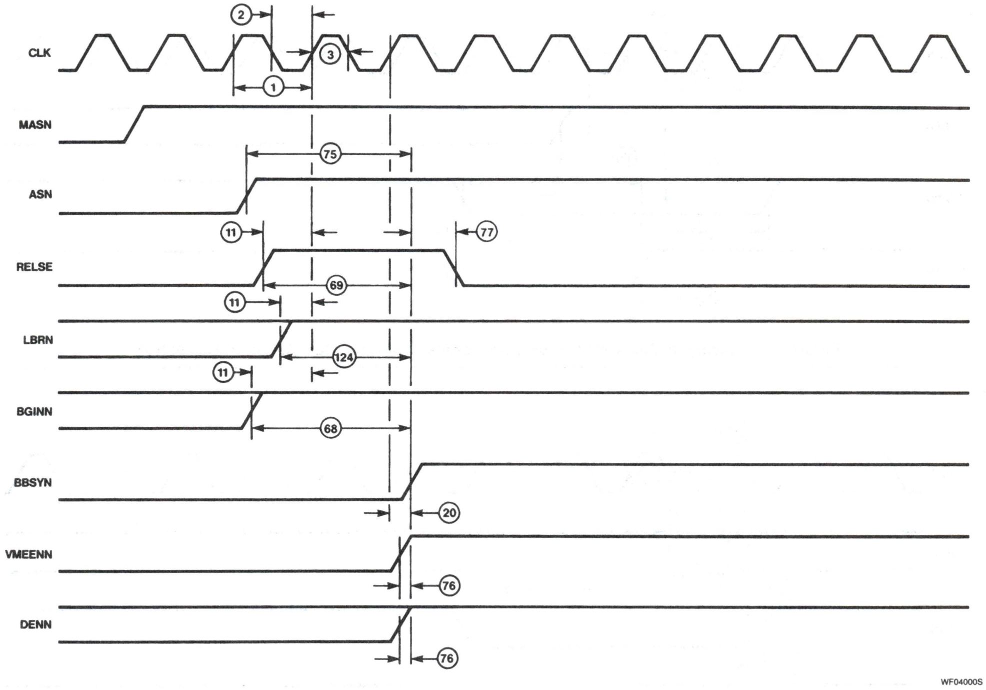

CLK
MASN
ASN
RELSE
LBRN
BGINN
BBSYN
VMEENN
DENN
②
③
①
(75)
(11)
(77)
(69)
(124)
(68)
(20)
(76)
WF04000S

Figure 15. Intercycle VMEbus Release

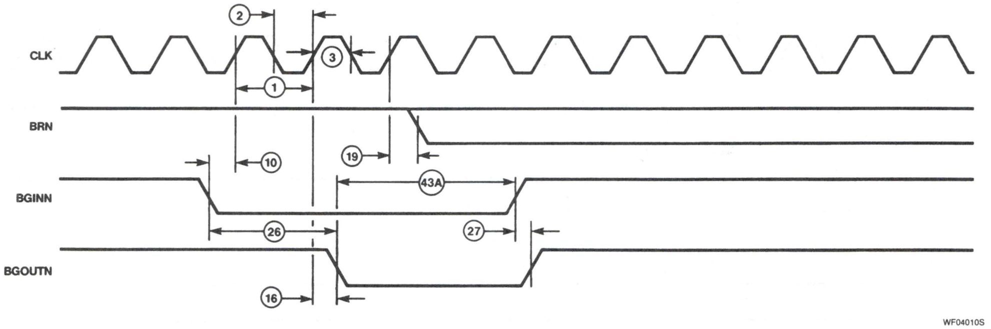

CLK
BRN
BGINN
BGOUTN
2
3
1
10
19
43A
26
27
16
WF04010S

Figure 16. Passing a Bus Grant, Including Earliest Possible Bus Request Thereafter

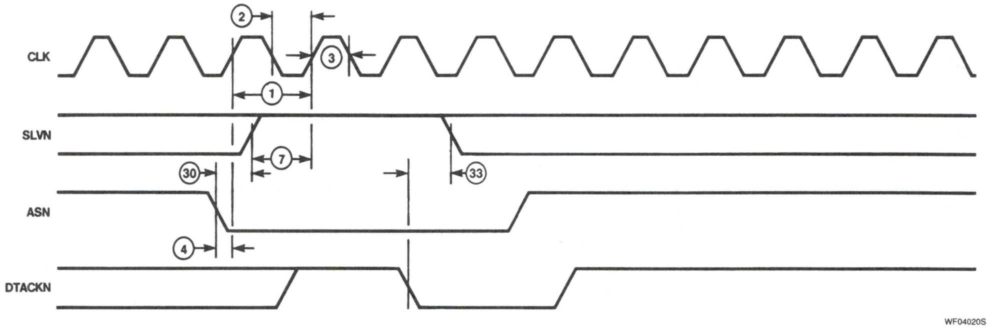

CLK
SLVN
ASN
DTACKN
2
3
1
7
30
33
4
WF04020S

Figure 17. Non-selecting Cycle on VMEbus

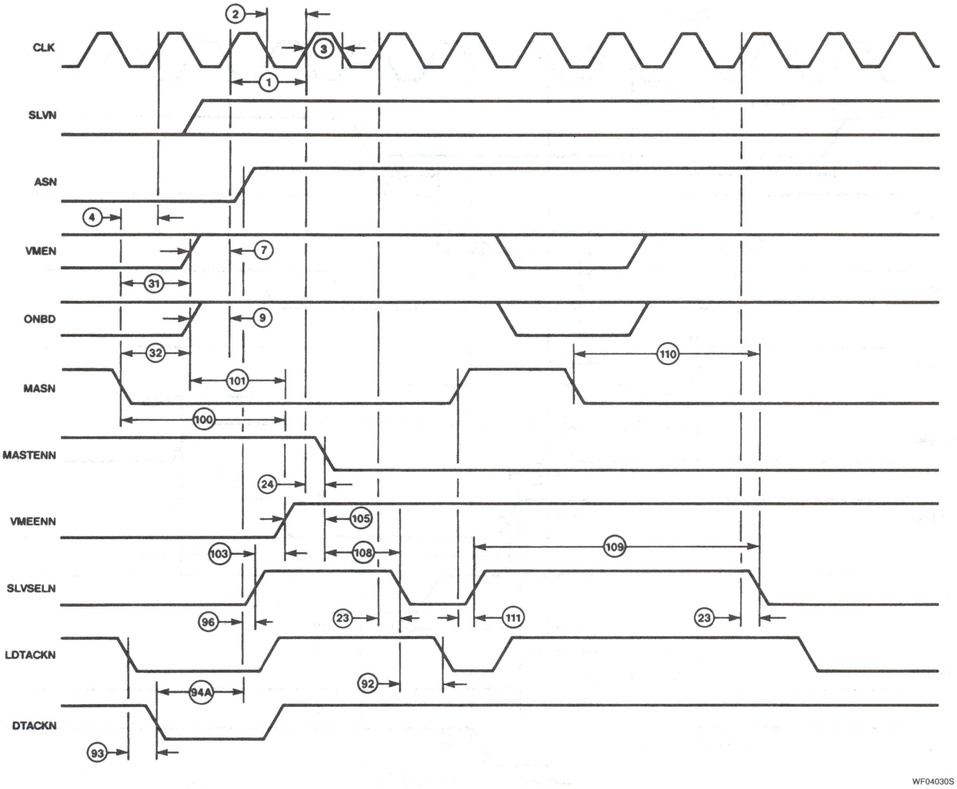

CLK
SLVN
ASN
VMEN
ONBD
MASN
MASTENN
VMEENN
SLVSELN
LDTACKN
DTACKN
1
2
3
4
7
9
23
24
31
32
92
93
94A
96
100
101
103
105
108
109
110
111
23
WF04030S

Figure 18. Two Onboard Cycles Following a Slave Cycle

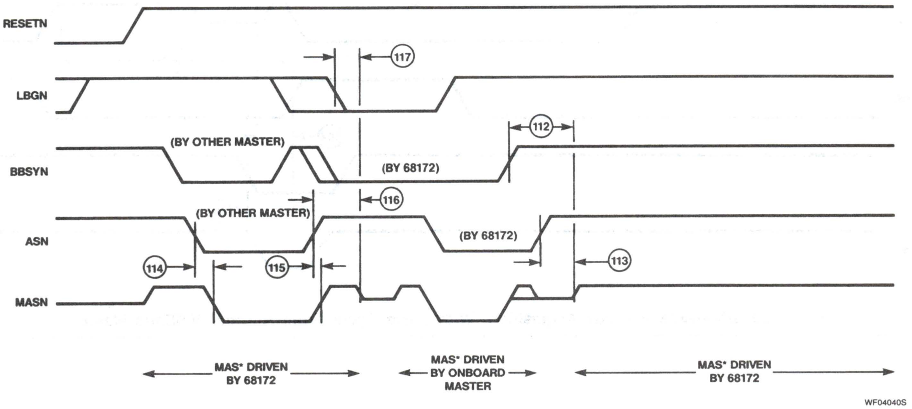

RESETN
LBGN
BBSYN
ASN
MASN
(BY OTHER MASTER)
(BY 68172)
(BY OTHER MASTER)
(BY 68172)
112
113
114
115
116
117
MAS* DRIVEN BY 68172
MAS* DRIVEN BY ONBOARD MASTER
MAS* DRIVEN BY 68172
WF04040S

Figure 19. AS* to MAS* Drive (DMA-type Operation)

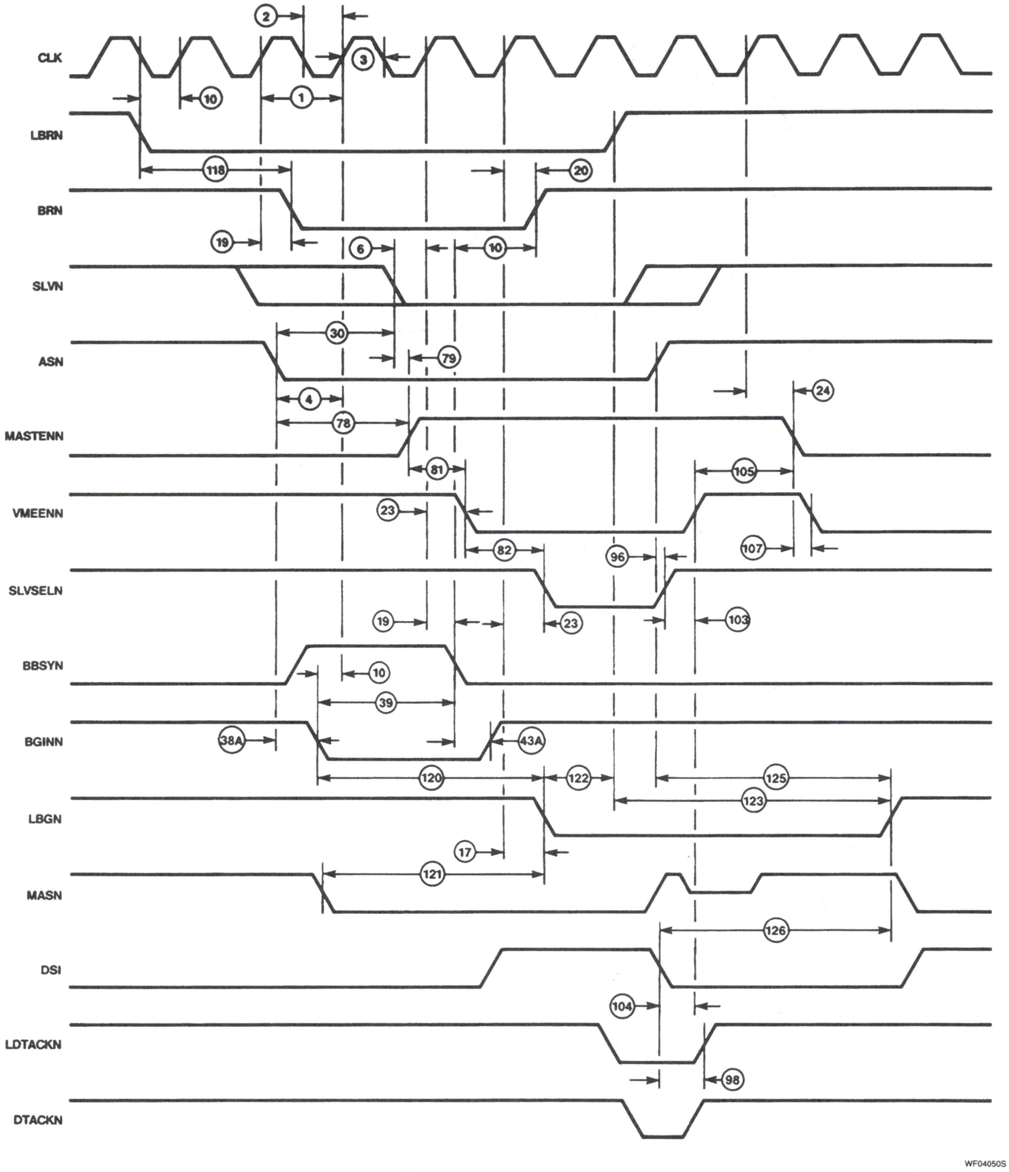

CLK
LBRN
BRN
SLVN
ASN
MASTENN
VMEENN
SLVSELN
BBSYN
BGINN
LBGN
MASN
DSI
LDTACKN
DTACKN
2
3
10
1
118
20
19
6
10
30
79
4
78
24
81
105
23
82
96
107
19
23
103
10
39
38A
43A
120
122
125
123
17
121
126
104
98
WF04050S

Figure 20. DMA-type VMEbus Acquisition, With Slave Cycle From Another VMEbus Master

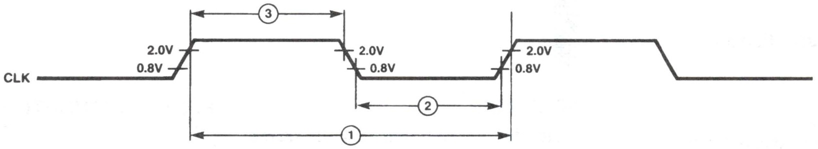

③
2.0V
0.8V
CLK
2.0V
0.8V
2.0V
0.8V
②
①
WF18720S

Figure 21. CLK Timing

Signetics

SCC68173
VMSbus Controller (VMSCON)

Preliminary Specification

Microprocessor Products

THE VMSbus

THE VMSbus provides a secondary control and data path in the backplane of VMEbus systems, and can also be used in other backplanes and in 'intercrate' (short inter-system) applications. Providing such capabilities as message passing, generalized event/interrupt communication, discrete signal control, and synchronization/semaphore operations, the VMSbus is especially significant for multiprocessor and/or fault-tolerant systems. With a 2.9Mbit/sec data rate in a backplane environment, collision-free self-arbitration during frame transmission, and variable message priority, the VMSbus offers a high-speed, deterministic path for the guaranteed delivery of short, urgent messages.

SCC68173 VMSbus CONTROLLER

The Signetics SCC68173 VMSbus Controller (VMSCON) is a register-oriented peripheral device that includes a collection of the basic functional modules described in the VMSbus specification. It interfaces to the VMSbus via the SCB68171 VMSbus Interface, and to a standard 68000 family bus or other parallel data buses. The SCC68173 includes the following VMSbus functions:

Qty	Function	Functional Modules
1	Controller	Header sender plus frame monitor
1	Talker	Data sender plus header receiver
1	Listener	Data receiver plus header receiver
4	Flags	Status flip-flop plus header receiver

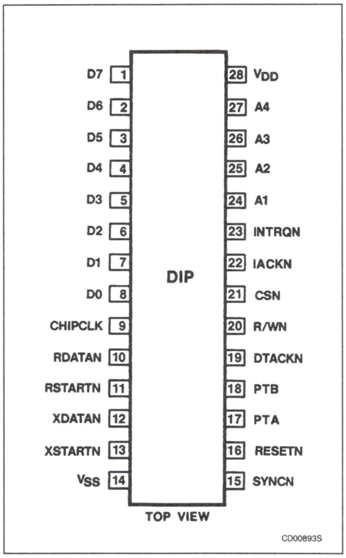
These functions are fully programmable so they can be used in a wide variety of combinations. The flags control two discrete output signal pins. All functions are designed to be driven via vectored interrupts.

FEATURES

- **Header sender paired with frame monitor**
- **Data sender with four-byte buffer**
- **Data receiver with four-byte buffer**
- **Four programmable flag modules**
- **Register blocks with sequential access**
- **Seven maskable interrupt sources**
- **Pending interrupt register**
- **Input signal can control frame transmission**
- **Interrupt vector base register**
- **Multiplexed header receivers**
- **Two direct control outputs**
- **68000 bus compatible**
- **CMOS Technology**
- **600 mil package width**

The SCC68173 is designed to be used with the SCB68171 which interfaces to the VMSbus clock (SERCLK) and data (SERDATN) lines. It also provides all the 68000 equivalent parallel bus controlling signals with an 8-bit bidirectional data bus. A set of command, status, and data registers are selected by the appropriate lower address bit lines.

PIN CONFIGURATION



VMSbus Controller (VMSCON) SCC68173

ORDERING INFORMATION

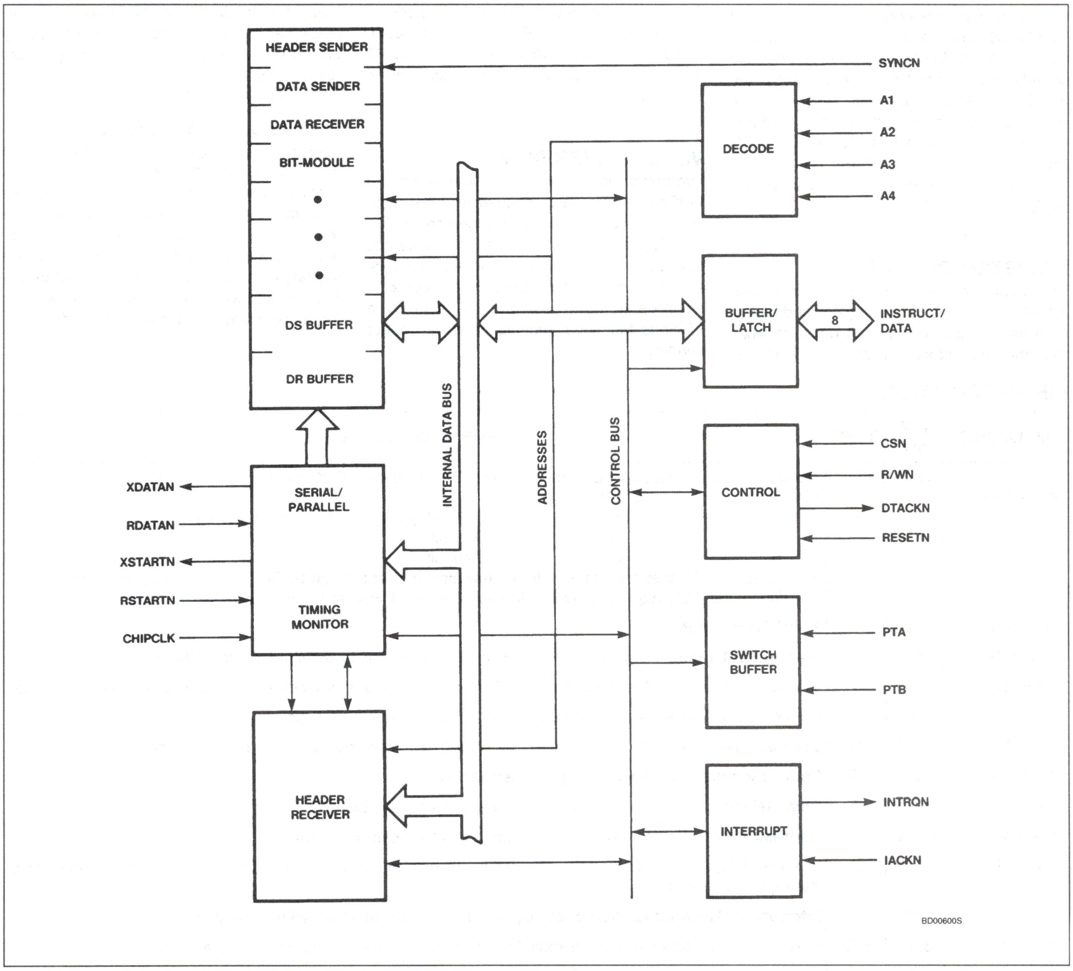
PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to 70°C
Ceramic	SCC68173C3I28
Plastic	SCC68173C3N28

BLOCK DIAGRAM

The SCC68173 VMSbus Controller consists of five major sections. The serial/parallel and parallel/serial conversion block controls all the timing of incoming and outgoing frames, does all on chip timing, and monitors the VMSbus. The header receiver block contains the programmed module addresses and incorporates the comparator logic to identify arriving S- and R-fields of a header subframe.

The register array can be loaded to control the header sender and the data sender, data receiver and bit modules.

Included in the array is a data sender buffer and a data receiver buffer. The interrupt control block provides request and acknowledge logic. The parallel bus interface includes four address lines, an 8-bit data bus, and bus control signals. There are also two direct control output lines available (PTA, PTB) .


HEADER SENDER
DATA SENDER
DATA RECEIVER
BIT-MODULE
DS BUFFER
DR BUFFER
SYNCN
A1
A2
A3
A4
DECODE
BUFFER/ LATCH
8
INSTRUCT/ DATA
INTERNAL DATA BUS
ADDRESSES
CONTROL BUS
CSN
R/WN
DTACKN
RESETN
CONTROL
XDATAN
RDATAN
XSTARTN
RSTARTN
CHIPCLK
SERIAL/ PARALLEL
TIMING MONITOR
SWITCH BUFFER
PTA
PTB
HEADER RECEIVER
INTERRUPT
INTRQN
IACKN
BD00600S

SCB68171 VMSbus INTERFACE

The VMSbus serial clock (SERCLK) provides for well-defined sampling times called S1 and S2. The Signetics SCB68171, a high speed bipolar interface, releases the VMSbus Controller from the VMSbus timing requirements. It also serves as a buffering device for multiple onboard VMSbus Controllers and provides the high current required on the SERDATN line. Figure 1 shows the VMSbus interface and controller interconnect. A timing diagram is shown in Figure 2.

As the SCC68173 sends data or prepares to do so, it also monitors the RDATAN and RSTARTN inputs for conflict with other transmitters. The XDATAN and XSTARTN outputs provide the signal timing required by the SCB68171. As an immediate response to a detected misplaced start bit, the VMSbus controller generates XSTARTN and XDATAN low simultaneously, which makes the SCB68171 drive SERDATN low immediately, to "jam" the VMSbus.

REGISTER BLOCKS

The SCC68173 has four address line inputs, A4 – A1, for a total of 16 register addresses. Of these, 10 are simple 8-bit registers while the other six address values (2, 3, 5, 10, 14, and 15) reference register blocks. Each register block has an internal address pointer that is used to access the "current" 8-bit register from among the several registers in the block. After the current register is read or written, the internal address pointer is incremented to point to the next register in the block. After the last register in a block is read or written, the pointer cycles back to the first register in the block.

The internal address registers, for blocks 2, 3, 5, and 10 are reset to the first register in the block, whenever any other register within R0 – R13 is accessed. The internal address pointers for R14 and R15 are independent, and are reset only when the DSE bit in R7 and the DRE bit in R8 (respectively) is changed.

VMSbus OVERVIEW

Information is transmitted in frames on the VMSbus. A frame is a sequence of consecutive bits in a prescribed arrangement or format. Figure 3 shows the three types of frames defined for the VMSbus. The first 26 bits, called the header subframe, are the same for all three frame types, and are sent by a VMSbus functional module called a header sender.

The header subframe contains two 10-bit fields called the S and R addresses. Either or both of these addresses may select one or more VMSbus functional modules called header receivers. Header receivers may be paired with other VMSbus functional modules called data senders and data receivers, and/or may control status bits (flipflops). The combined functions are called talkers, listeners, and flags, respectively.

The first frame in Figure 3 shows the case where one (or more) of the header receivers selected (by either the S or R address) is not ready for the frame. In this case, the not-ready header receiver cancels the frame by driving the value 111 in the frame type field. Such cancelled frames end immediately after the frame type field.

The second frame in Figure 3 shows a non-data frame. This type of frame results when the S address does not select a talker. In normal operation, the R address of a non-data frame does not select a listener. The frame type field contains zero and is immediately followed by the frame status field, which indicates whether flags were selected by the S and R addresses respectively, and whether there were any problems with the frame.

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
XDATAN, XSTARTN	12,13	O	**Transmit Data Control:** Open drain outputs. Set up during CLK low time as follows: XDATAN XSTARTN H H Zero bit (or not transmitting a frame) L H One bit H L Start bit Also, if RSTARTN goes low during a frame (misplaced start bit), the SCC68173 drives both of these signals low during the CLK high time, which forces SERDATN low and jams the bus.
RDATAN	10	I	**Serial Data Input**
RSTARTN	11	I	**Start Detect Input:** Active low whenever a start condition is detected on the VMSbus.
CHIPCLK	9	I	**Clock:** The CLK signal is derived from SERCLK – the serial bus clock – by the bipolar interfacing device.
RESETN	16	I	**Reset:** Active low signal, which clears most of the controller's logic.
PTA, PTB	17,18	O	**Control:** These outputs are controlled by on-chip bit modules for direct control purposes.
DTACKN	19	O	**Data Transfer Acknowledge:** Open drain, active low.
CSN	21	I	**Chip Select:** This active low signal activates transfer on D0 – D7.
A1 – A4	24 – 27	I	**Address Lines:** These lower address lines select on-chip registers.
INTRQN	23	O	**Interrupt Request:** Active low, open drain output which signals the CPU that one or more of seven maskable interrupting conditions is true.
IACKN	22	I	**Interrupt Acknowledge:** Active low input indicating an interrupt acknowledge cycle.
D0 – D7	8 – 1	I/O	**Data Bus:** Bidirectional 3-State data bus used to transfer commands, data, and status.
R/WN	20	I	**Read/Write:** Read/write control.
SYNCN	15	I	**Synchronize:** Input which controls frame transmission by the header sender module.
V_{DD}	28	I	**+ 5V ±5% power input**
V_{SS}	14	I	**Ground**

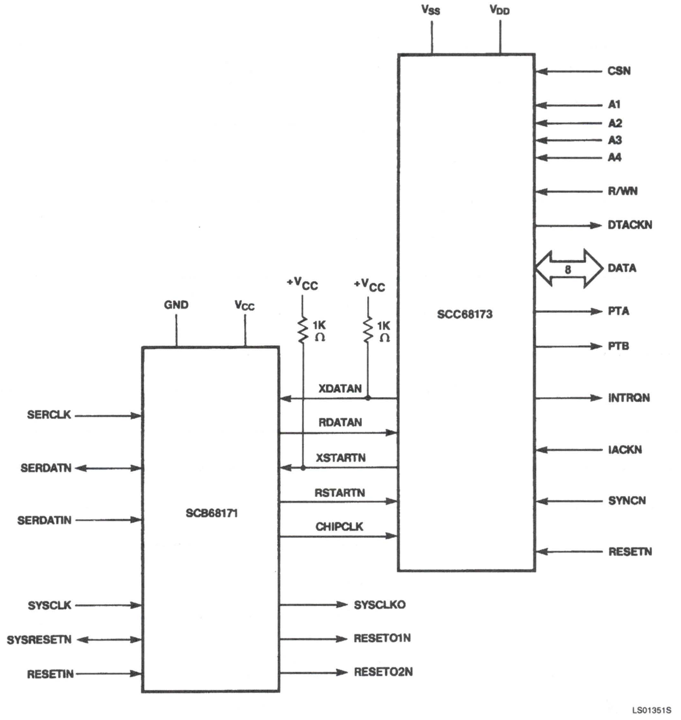


Figure 1. VMSbus Interface and Controller Interconnect

The third and last frame in Figure 3 shows a data frame. In this case, the S address selected a talker function. The data sender sends one of the values 001 – 110 in the frame type field, indicating the number of data bytes which follow. In normal operation, the R address selects a listener function, which receives the data. The data is then followed by the frame status field, which indicates if there were any problems with the frame.

As shown in Figure 3, every frame is followed by a single bit called a jam bit. The jam bit should always be high/zero. This forms a guard against desynchronization among various VMSbus modules, as to where frames begin and end.

All frames begin with a special unique start bit. Each frame monitor module tracks every VMSbus frame – if it detects a start bit within a frame, it jams the VMSbus with a string of 512 low/one bits. Since the longest VMSbus frame is 287 bits, this jam sequence is sure to affect the jam bit of any frame currently in progress.

If any VMSbus module detects a jam bit low/one, it ignores the preceding frame. The jam sequence also serves to resynchronize all VMSbus modules, which wait until it is over before transmitting any more frames.

Since a VMSbus typically includes a number of header senders, there must be a way to control when they start sending frames. As already noted, frame monitor modules track the progress of every VMSbus frame. If a header sender is signaled to start a frame while its paired frame monitor is tracking a frame, it waits until the current frame is completed.

VMSbus data is wire-ORed, so that if several modules send data in the same bit time, the result is the logical OR of their data. This fact is used to provide bus arbitration as frames are sent. Whenever a module sends a zero/high bit, it also senses the result bit on the bus. If it senses a one/low, it has lost the arbitration and stops sending data. Typically, a header sender waits until the current frame is over, and then tries to send its header subframe once again.

This arbitration method ensures that data is never lost or garbled because of contention for use of the VMSbus. If several modules send data, the data with the highest binary value gets through without distortion. Arbitration is always used by header senders; its use is optional for data senders, depending on the DSAE bit in the header subframe.

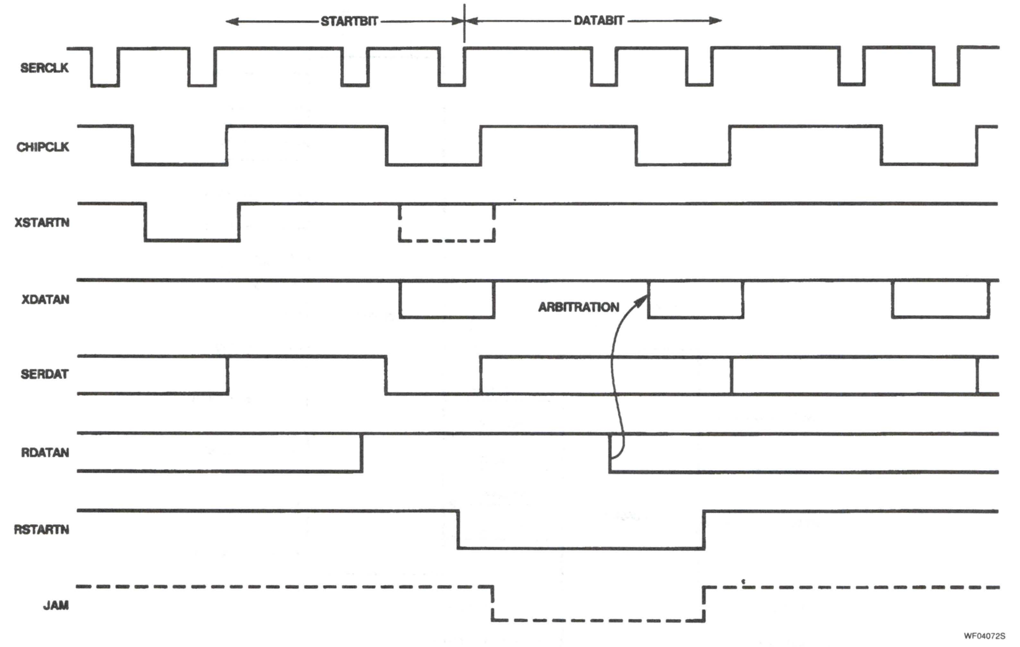


Figure 2. Timing

CONTROLLER FUNCTION (HEADER SENDER/FRAME MONITOR)

Before a header subframe can be sent, the information in it must be programmed into the SCC68173. This includes the priority field (in register R0), the S and R addresses (register blocks R2 and R3), and the DSAE bit (in register R0).

If no automatic retries should be done in response to less of VMSbus arbitration, the RTE bit in R0 should be written as 0. If up to 1 – 14, automatic retries should be done, then RTE should be written as 1, and the number of retries should be written into R1. If up to 15 automatic retries should be done, then RTE should be written as 1, but R1 need not be written.

If XRS in R0 is zero, then the S address is taken from register block R2 and the R address is taken from R3. If XRS is one, then the S address is taken from R3 and the R address from R2. This feature can be useful, for example, to send set and reset frames without reloading R2 and R3.

When the necessary information is loaded, the SCC68173 can be commanded to send the header subframe by software (setting the HSA bit in R0) or by its onboard hardware (SYNCN input low after software has set the SCE bit in R0). After any current frame in progress has completed, the controller function then tries to send the header subframe. This attempt may have one of several results:

1. The Controller may lose VMSbus arbitration while trying to send the header subframe. In this case it always sets the LAB bit in R1. Its other actions depend on the RTE bit in R0 and on the retry counter which can be loaded by writing R1:
   a. If RTE is zero, the SCC68173 clears the HSA bit in R0. Note that in any case where the SCC68173 clears the HSA bit in R0, it stops trying to send the header subframe, and the header sender becomes inactive. Also, if HS in R13 is set, then it sets HS in R12 and requests interrupt on IRQN.
   b. If RTE is one, the SCC68173 decrements the 4-bit retry counter. If this operation does not result in a borrow, the controller function waits for the current frame to complete, and then tries to send the header subframe again.
   c. If RTE is set, and decrementing the retry counter results in a borrow, the controller function clears the HSA bit in R0.
2. If the header sender wins the VMSbus arbitration through sending the HSVAL bit as low/one, it clears the LAB bit in R1 and stops sending. The frame monitor continues to track the resulting frame, through the final jam bit. If the jam bit is "one", the controller sets the jam bit in R1 and clears the HSA bit in R0. (Jam should be the overriding result, and should always be reported by the controller hardware. If this result does not preclude other bits in R1, at least jam should be the first thing checked by the software.)
3. The resulting frame may be cancelled. In this case the SCC68173 sets the CNC bit in R1 and clears the HSA bit in R0.
4. A non-data frame may result. In this case SCC68173 sets the SNT bit in R1, clears the DFR bit in R1, saves the frame status in R1 and clears the HSA bit in R0.
5. A data frame may result. SCC68173 action in this case is the same as for a non-data frame, except that it sets the DFR bit in R1.

In either case 4 or 5 above, software should check the DFR and frame status bits in R1 to see that the expected set of header receivers was selected and that the frame was otherwise correct. For cases 1 and 3 (lost arbitration and cancelled frame), the frame status

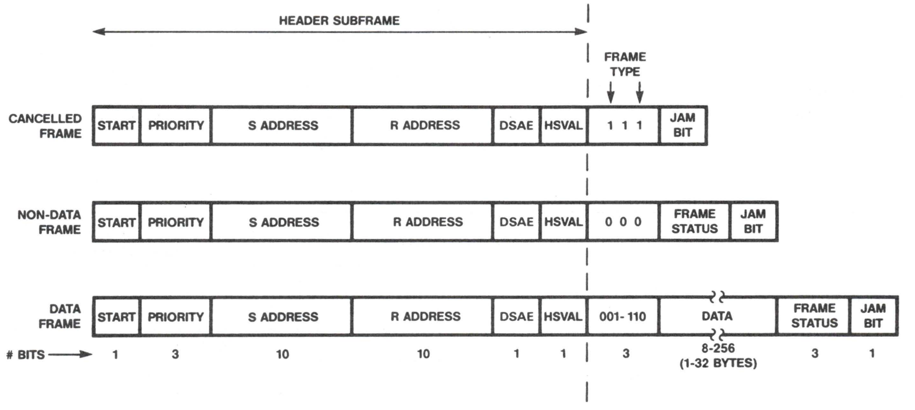


Figure 3. VMSbus Frames

field contains the monitored priority field from the bus, rather than the frame status field.

The priority, DSAE, XRS, and RTE bits can be set in R0 by the same command which sets HSA or SCE. When the SCC68173 clears HSA in R0, it does not clear SCE. To prevent additional transmissions in case SYNCN is still low, the SCC68173 requires that SYNCN go high again before it is enabled to start another frame.

System software can abort a series of retries by rewriting R0 with HSA cleared. It should then check HSA in R0 until it is zero, because if HSA is rewritten while a VMSbus frame is in progress, the new state does not become effective until the end of the frame.

HEADER RECEIVERS

There are six header receivers in the SCC68173, called HR0 through HR5. Each of them can compare a 10-bit address, assigned to it by system software, to the S and R addresses in frames on the VMSbus. When an address matches, a header receiver signals its associated module that it is selected by the frame. It is important to note that header receivers and their associated modules respond identically to VMSbus frames that are initiated by their on-chip header sender, and to frames initiated by other header senders.

Addresses are assigned to header receivers by writing register block R5. The first (reset) byte in register block R5 corresponds to the less-significant 8 bits of HR0's address, while the last (12th) byte in R5 corresponds to the 2 high-order bits of HR5's address.

Note that R5 is a write-only register block. This means that if the address assigned to a particular header receiver is to be changed, all the addresses, up to and including the desired one, must be rewritten. In general this implies that system software must maintain a memory table containing the image of register block R5.

The six header receivers select specific modules in the SCC68173 as controlled by register R6. The selection coding is shown in Table 2. While each header receiver selects one module, the same address can be programmed into more than one header receiver. Each module may be selected by 0, 1, 2, or 3 header receivers.

Register R4 controls which header receivers are enabled. A module can only respond to VMSbus frames if one or more header receivers (that are enabled in R4) are programmed to select it, in R6.

A header receiver should be disabled in R4 before its selection criteria are changed in R5 and/or R6.

The six modules that are selected by the header receivers are one data sender, one data receiver, and four flags. They are described separately in the following sections. The data sender can be selected only by a match in the S address, while the data receiver can be selected only by a match in the R address. Flags can be selected by a match in either address.

DATA SENDER

The function of this module is to send data on the VMSbus (to a data receiver module) when it is selected by the S address in a frame. If the selected data sender is on the same board (in the same SCC68173) as the header sender that initiated the frame, the overall VMSbus operation is called a write frame. If the selected data receiver is on the same board (in the same SCC68173) as the initiating header sender, the operation is a read frame. If both the data sender and data receiver are on different boards (in different SCC68173's), the operation is called a move.

System software should set up a data sender for transmission as follows:

1. If necessary, program the address(es) for the Data Sender into one or more Header Receiver(s), by writing R5.
2. If necessary, establish the link between the Data Sender and one or more Header Receiver(s), by writing R6.
3. Enable the linked Header Receiver(s), by writing R4.
4. If necessary, enable the data sender interrupt in register R13.
5. Write the data to be sent into register block R14.
6. Write register R8 with the frame type code for 1, 2, or 4 bytes, plus the DSE bit and optionally the DSM bit (see the discussion that follows on writing R8).
7. For a write frame, program the header sender to initiate a frame with the data sender's address in the S field.

Because the DSE bit in R8 can be set and cleared by system software and also cleared by a VMSbus frame, a special technique must be used by system software in writing R8. For bits DSE, DSM, and DST, writing a '1' in each bit position indicates that the bit should be

Table 1. Registers

Header sender control and monitor	R0	R/W	Priority			DSAE	XRS	RTE	SCE	HSA
	R1	R	SNT	CNC	LAB	JAM	DFR	Priority/Status		
		W	Reserved				Preset retry cntr			
	R2*	R/W	S-FIELD							LSB
			Not Used						MSB	
	R3*	R/W	R-FIELD							LSB
			Not Used						MSB	
Module addressing	R4	R/W	Reserved	Reserved	HR5	HR4	HR3	HR2	HR1	HR0
					Header receiver enable					
	R5*	W	HR0 address							LSB
			Not Used						MSB	
			HR1–5 addresses							
	R6	R/W	Header receiver mux							
Module control	R7	R/W	Write Data	DRE	DRM	DRV	DRA	DR frame type		
				Write select						
	R8	R/W	Write Data	DSE	DSM	DST	Write Enable	DS frame type		
				Write select						
	R9	R/W	Write Data	Reserved			FF3	FF2	FF1	FF0
							Write select			
	R10*	R/W	Connect FF1	FF1 mode			Connect FF0	FF0 mode		
			Connect FF3	FF3 mode			Connect FF2	FF2 mode		
Interrupt control	R11	R/W	Interrupt vector (lower 3 bits read only)							
	R12	R/W	INP	HS	DR	DS	FF3	FF2	FF1	FF0
				Pending interrupts (read), reset mask (write)						
	R13	R/W	INE	HS	DR	DS	FF3	FF2	FF1	FF0
				Interrupt enable						
Buffers	R14*	R/W	Data sender buffer							
	R15*	R	Data receiver buffer							

NOTE:
*Indicates sequentially-accessed register block.

written with the data (0 or 1) contained in the high-order bit (labeled Write Data in Table 1). Also, the data sender frame type bits are written from the three least significant bits only if the next more significant bit is a '1' (labeled Write Enable in Table 1).

When the data sender is selected by the S address in a header subframe, it begins operation by sending a code in the frame type field which directly follows the header subframe. If bit DSE in register R8 is zero, indicating that system software has not set up current data and enabled the data sender, it sends 111 in the frame type field, cancelling the frame.

If DSE in R8 is one, the data sender sends one of the values 001, 010, or 011 in the frame type field, as controlled by the low order bits in R8. These values indicate 1, 2, or 4 bytes of data, respectively, following the frame type field. (If one of the illegal values 000, 100 – 111, is inadvertently programmed into the three low-order bits of R8, the SCC68173 sends 111 in the frame type, cancelling the frame.)

While sending the frame type, the data sender uses VMSbus arbitration as described previously in VMSbus Overview. If it loses the arbitration, it stops sending and captures the winning frame type value. If the value is 111, another module (typically the data receiver) is not ready for the frame and has cancelled it. In this case the data sender ignores the frame, keeping DSE set.

If the data sender loses the arbitration to a frame type value 010 – 110, then another data sender is also selected by the S address, and it is set up to send more data than is this data sender. In this case, this data sender waits out the number of bits indicated by the winning frame type value, and then sends the error code 110 in the frame status field.

If the data sender wins the VMSbus arbitration in the frame type field, it goes on to send 1, 2, or 4 bytes of data from register block R14. If the DSAE bit in the initiating header subframe was 1, it uses VMSbus arbitration while sending the data, and if it loses the arbitration it ignores the frame, leaving DSE set. If DSAE was 0, it does not use VMSbus arbitration, but simply places each data bit on the VMSbus.

After the data, the data sender sends the code 010 in the frame status. This OR's with the code 001 from the data receiver to make the "success" code 011.

The data sender also samples the frame status and the following jam bit. If the status code is 011 and the jam bit is 0, then it sets the DST bit in register R8. Also, if the DSM bit in R8 is zero, it clears the DSE bit in R8 and, if the INE and DS bits in R13 are both one, it requests interrupt on IRQN. If DSM is one, the SCC68173 leaves DSE set and does not request interrupt (regardless of INE and DS in R13).

The DSM bit in R8 thus controls the gross operational state of the data sender. When DSM is zero, setting DSE enables the module to send data once, then clear DSE and cancel any further frames that select it, until system software (loads new data and) sets DSE again. When DSM is one, setting DSE enables the data sender to send the data in register block R14 as many times as it is selected by VMSbus frames, until system software clears DSE.

Software cannot access register block R14 while DSE is set. This restriction applies to both reading and writing, and can be handled in one of three ways:

1. Wait until DSE is cleared by the SCC68173. This is normal operation with DSM zero, but is not applicable if DSM is one.
2. Rewrite R8 to make DSE zero. Software should then check R8 until DSE reads back as zero, because if the data sender is selected by the current VMSbus frame, the clearing of DSE does not become effective until the end of the frame.
3. Simply go ahead and access register block R14. This automatically clears DSE, but the entire access operation is delayed if the data sender is selected by the current VMSbus frame, until the end of the frame. Systems which cannot tolerate a DTACKN delay of up to 270 bit times should use method 2 above. (Since SCC68173 data senders are restricted to 1 – 4 byte data, the practical maximum delay is about 40 bit times except in the case where there is a data size conflict between a SCC68173 and some other kind of VMSbus data sender. 40 bit times is about 14μsec at a 2.9MHz data rate.)

Table 2. Header Receiver Multiplexing in Register A6

A6 BITS	HR	VALUE	MODULE SELECTED	BY MATCH IN
7 (MSB)	3	0	Bit module FF3	S, R
		1	Data receiver	R
6	5	0	Data receiver	R
		1	Bit module FF2	S, R
5, 4	2	00	Bit module FF2	S, R
		01	Bit module FF1	S, R
		10	Data receiver	R
		11	Bit module FF2	S, R
3, 2	1	00	Bit module FF1	S, R
		01	Data sender	S
		10	Bit module FF2	S, R
		11	Bit module FF1	S, R
1	4	0	Data sender	S
		1	Bit module FF1	S, R
0 (LSB)	0	0	Bit module FF0	S, R
		1	Data sender	S

DATA RECEIVER

A data receiver captures data from VMSbus frames in which it is selected by the R address. System software should set up a data receiver as follows:

1. If necessary, program the address(es) for the Data Receivers into one or more Header Receiver(s), by writing R5.
2. If necessary, establish the link between the Data Receivers and one or more Header Receiver(s), by writing R6.
3. Enable the linked Header Receiver(s), by writing R4.
4. If necessary, enable the DR interrupt in register R13.
5. Write register R7 with DRE one, and optionally, DRM one.
6. For a read frame, program the header sender to initiate a frame with the data receiver's address in the R field.

Because the DRE and DRA bits in R7 can be set and cleared by system software and also cleared by a VMSbus frame, a special technique must be used by system software in writing R7. For bits DRE, DRM, DRV, and DRA, a '1' in that bit position indicates that the bit should be written with the data contained in the high-order bit (labeled Write Data, in Table 1).

When the data receiver is selected by the R address in a header subframe, it clears the DRA bit in R7. If bit DRE in R7 is zero, it sends 111 in the frame type field that follows the header subframe, thereby cancelling the frame.

If DRE in R7 is one, the data receiver samples the frame type field and saves the result in the low-order 3 bits of R7. If the frame type is 111, the frame has been cancelled by another VMSbus module (typically a data sender). In this case the data receiver ignores the frame.

If the frame type field is 000, the data receiver then sends the error code 101 in the immediately following frame status field.

If the frame type field is 100, 101, or 110, the data receiver waits for 64, 128, or 256 bits of data (respectively) to go by, and then sends the error code 101 in the frame status field.

If the frame type field is 001, 010, or 011, the data receiver captures the following 1, 2, or 4 data bytes (respectively), and saves them in register block R15. It then sends the frame status code 001, which OR's with the code 010 from the data sender to produce the "successful" status code 011.

The data receiver also samples the frame status and the following jam bit. If the status code is 011 and the jam bit is 0, then it sets the DRV bit in R7. Also, if the DRM bit in R7 is zero, the data receiver clears the DRE bit in R7 and, if the INE and DR bits in R13 are both one, it requests interrupt on the IRQN pin. If DRM is one, it leaves DRE set and does not request interrupt, regardless of INE and DR in R13.

Thus the DRM bit governs the gross operational mode of the data receiver. If DRM is zero, once the data receiver has successfully received data, it clears DRE and cancels any further frames addressed to it, until system software (reads the data from R15 and) sets DRE again. When DRM is one, the data receiver will accept data into R15 from any frame addressed to it.

When DRM is one, system software must guard against the possibility that a new VMSbus frame could be received while it is reading data out of register block R15, which would result in reading out a mixture of old and new data. It can do this in one of two ways:

1. Rewrite R7 with DRE zero. Software should then check R7 until DRE reads

Table 3. Operating Modes for Flags in A10

VALUE	MODULE TYPE	INTERRUPT ON
000	Disabled (Cancellation)	
001	Simple flag	Toggle
010	Simple flag	Set
011	Simple flag	Clear
100	Semaphore	Set
101	Semaphore	Clear
110	Token	Set
111	Token	Clear

Table 4

FF1 TO PTA OR FF3 TO PTB	FF0 TO PTA OR FF2 TO PTB	FF1 OR FF3	FF0 OR FF2	PTA OR PTB
0	0	X	X	Low
0	1	X	0	Low
0	1	X	1	High
1	0	0	X	Low
1	0	1	X	High
1	1	0	0	Low
1	1	0	1	High
1	1	1	0	High
1	1	1	1	High

back as zero, because if the data receiver is selected by the current VMSbus frame, the clearing of DRE does not become effective until the end of the frame. When DRE is zero, the software can safely read out register block R15. Typically, it then sets DRE again. This method has the disadvantage that it results in cancellation of any frame which selects the data receiver while its data is being read out.

2. Set the DRA bit in register R7, read out register block R15, then read R7. If the DRA bit in R7 is still set, the data from R15 and the frame type code in R7 are valid. If DRA has been cleared (by the SCC68173 because it started receiving a frame), then the software should set DRA and try again.

FLAGS

Flags essentially implement a flip-flop or latch which is set when the module is selected by the S address in a frame, and reset when the module is selected by the R address in a frame. Four flags are provided in the SCC68173, called FF0 through FF3. These modules can also be directly controlled by system software, and can be programmed to control two discrete SCC68173 output pins, PTA and PTB.

Register R9 provides direct sensing and control of the state of the flags. Since the state of the modules can be changed both by system software and by VMSbus frames, a special technique must be used in writing R9. A '1' in any or all of the four least significant bit positions indicates that a flag should be written with the data contained in the high-order bit (labeled Write Data in table 1).

Register block R10 provides control of the operation of each flag. In R10, there are four bits for each module, of which the high-order bit determines whether the module is connected to the PTA or PTB pin, and the three less significant bits determine the module's operational mode, as shown in Table 3.

If a flag is programmed with the 000 code and is also connected to a header receiver (in R6) that is enabled (in R4), then any frame that selects the flag (in either the S or R fields) will be cancelled.

The simple flag, semaphore, and token categories also refer to circumstances where flags cancel frames that are addressed to them.

A simple flag does not cancel frames based on its state. That is, it accepts "set frames" even if it is already set, and accepts "reset frames" if it is already reset. On the other hand, if the module's interrupt was enabled (in R13) at the time of a previous frame, and the interrupt is still pending, then any frame that selects the flag (in either the S or R fields) will be cancelled.

A semaphore module cancels a set frame if it is already set, but accepts a reset frame if it is already reset. It does not cancel frames based on its pending interrupt status.

A token module cancels a set frame if it is already set, and cancels a reset frame if it is already reset. It does not cancel frames based on its pending interrupt status.

Within each category there is a choice as to when the module requests interrupt. This choice is significant only if the module's interrupt enable bit is set in R13. System software should set up a flag as follows:

1. If necessary, disable the associated header receiver(s) in R4.
2. If necessary, program the initial state of the module, in R9.
3. Program the module's operating mode, in R10.
4. If necessary, clear any pending interrupt for the module, in R12.
5. If necessary, enable interrupt for the module, in R13.
6. Program the module's address via R5.
7. Connect a header receiver to the module, in R6.
8. Enable the header receiver in R4.

The PTA and PTB pins are high-active, and go low on reset. Each pin can reflect the state of 1 or 2 flags; as shown in Table 4.

INTERRUPTS

As described in the previous sections, the SCC68173 can be set up to generate an interrupt request based on any of seven events:

1. When the header sender finishes sending a header subframe, and clears HSA.

2. When the data sender sends data in a frame, and clears DSE.

3. When the data receiver receives data in a frame, and clears DRE.

4–7. When flag FF0–FF3 is set and/or reset by a frame.

Each of these seven potential interrupt sources is conditioned on a corresponding bit in register R13. When an R13 bit is one, the corresponding module's interrupt is enabled. The most significant bit in R13, INE, controls the overall interrupt enable of the SCC68173.

Register R12 contains a corresponding set of interrupt pending bits. An interrupt pending bit is set if the appropriate VMSbus event occurs, and the module's interrupt enable bit is one in R13. The most significant bit of R12 is the OR of the other seven bits.

A pending bit in R12 can be directly reset by writing a 1 to it. Disabling an interrupt channel in R13 automatically resets the corresponding pending bit. When an interrupt service routine processes a SCC68173 interrupt, it should clear one or more of the pending bits and/or enable bits and/or make INE zero before dismissing the interrupt, to prevent a second interrupt resulting from the same event.

When the SCC68173 detects an interrupt acknowledge cycle (IACKN low) and it is requesting interrupt on IRQN (or is about to do so), it responds by placing an interrupt vector on the D7 – D0 lines and asserting DTACKN. Typically, this vector has its high-order 5 bits taken from register R11, and its low-order 3 bits in accordance with the highest-priority pending interrupt:

110	Header sender (highest priority)
101	Data receiver
100	Data sender
011	Flag FF3
010	Flag FF2
001	Flag FF1
000	Flag FF0 (lowest priority)

The exception to this vector structure is where R11 has not been programmed since the SCC68173 was reset. In this case, the vector hexadecimal 0F is returned regardless of which modules have interrupt pending.

Jam Sequence

As described in VMSbus Overview, the SCC68173 monitors the VMSbus at all times (other than when RESETN is low), tracking the entire length of each frame and checking for start bits within each frame. If such a start bit is detected, it means that this SCC68173 is out of frame synchronization with one or more other VMSbus devices. In this case, this SCC68173 must send a string of 512 one bits to resynchronize the VMSbus, starting with the bit after the misplaced start bit. The timing of the SCC68173/SCB68171 interface does not allow this to be done in the normal (synchronous finite-state machine) way. Instead, if RSTARTN goes low from the SCB68171 during a frame, this signal makes the SCC68173 produce XSTARTN and XDATAN low combinatorially. This combination makes the SCB68171 asynchronously drive SERDATN low for a one-bit. By the next bit time, the SCC68173 is set up to send the rest of the jam sequence in the normal way. A jam sequence maintains all SCC68173 hardware as it was at the start of the frame.

Reset

A low on the RESETN input clears the following bits to zero:

HSA and SCE in register R0

DRE and DRV in R7

DSE and DST in R8

All bits in registers R1, R4, R6, R12, R13

It also sets register R11, the interrupt vector, to the value hexadecimal 0F.

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
V_{CC}	DC supply voltage	−0.5	+7	V	
$\pm I_O$	DC output source or sink current – standard outputs		25	mA	for $-0.5V < V_O < V_{CC}+0.5V$
$\pm I_{CC}$; $\pm I_{GND}$	DC V_{CC} or GND current for types with: – standard outputs		50	mA	

OPERATING CONDITIONS

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
V_{CC}	DC supply voltage	4.5	5	5.5	V
V_I	DC input voltage	0		V_{CC}	V
V_O	DC output voltage	0		V_{CC}	V
T_{amb}	Operating ambient temperature range	0		70	°C

DC ELECTRICAL CHARACTERISTICS

SYMBOL	PARAMETER	T_{amb}(°C)					UNIT	TEST CONDITIONS		
		74HCT						V_{CC} V	V_I	Other
		+25			-40 to +85					
		Min	Typ	Max	Min	Max				
V_{IH}	High level input voltage	2.0	1.6		2.0		V	4.5 to 5.5		
V_{IL}	Low level input voltage		1.2	0.8		0.8	V	4.5 to 5.5		
V_{OH}	High level output voltage standard outputs	3.98	4.32		3.84		V	4.5	V_{IH} or V_{IL}	$-I_O = 4.0mA$
V_{OL}	Low level output voltage standard outputs		0.15	0.26		0.33	V	4.5	V_{IH} or V_{IL}	$I_O = 4.0mA$

AC ELECTRICAL CHARACTERISTICS

NO.	CHARACTERISTIC	Tentative MIN	Tentative MAX	Tentative UNIT
1	A1-4 setup time to CSN low	0		ns
2	A1-4 hold time from DTACKN low	100		ns
3	RWN setup time to CSN low	0		ns
4	RWN hold time from CSN high	0		ns
5	DTACKN low to CSN high	0		ns
6	CSN high	100		ns
7	Read data valid to DTACKN low	0		ns
8	Read data to float from CSN high	0	100	ns
9	Write data setup time	0		ns
10	Write data hold time from DTACKN low	25		ns
11	CSN low to DTACK low	100+2 × ⑮		ns
12	DTACKN fall time		25	ns
13	CSN high to DTACKN high	0		ns
14	DTACKN low to PTA, PTB valid		100	ns
15	CLK period	320		ns
16	CLK high time	200		ns
17	CLK low time	100		ns
18	CLK rise time		6	ns
19	CLK fall time		6	ns
20	CLK high to XSTARTN low		150	ns
21	CLK low to XSTARTN off	5	50	ns
22	CLK low to XDATAN low		90	ns
23	CLK high to XDATAN off	5	80	ns
24	RSTARTN low to CLK high		−30	ns
25	RSTARTN hold time from CLK low	50		ns
26	RSTARTN high to CLK high	0		ns
27	RDATAN valid to CLK low		0	ns
28	RDATAN hold time from CLK high	100		ns
29	Misplaced RSTARTN low to XSTARTN low		100	ns
30	Misplaced RSTARTN low to XDATAN low		100	ns
31	RESETN low time	10 × ⑮		ns
32	INTRQN high to DTACK low	0		ns
33	Read data valid to DTACKN low	0		ns
34	Read data to float from IACKN high	0	100	ns
35	IACKN high to DTACKN high	0	100	ns
36	SYNCN pulse low time	100		ns
37	Frame end to INTREQN low		100 + ⑮	
38	Frame end to PTA, PTB valid		100 + 2 × ⑮	

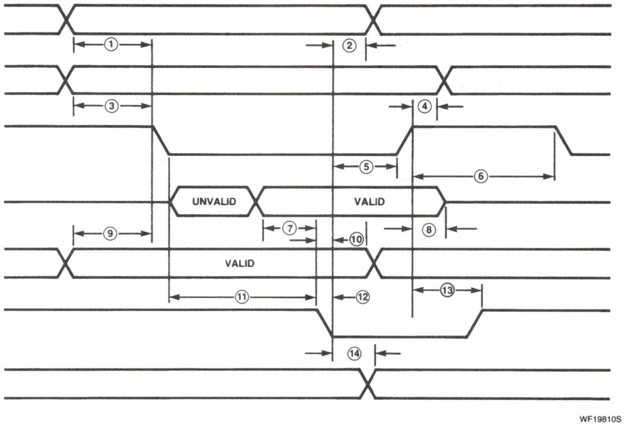

A1–4
①
②
RWN
③
④
CSN
($\overline{AS}$, $\overline{DS}$ LOW)
⑤
⑥
D0–7
(READ)
UNVALID
VALID
⑦
⑧
⑨
⑩
D0–7
(WRITE)
VALID
⑪
⑫
⑬
DTACKN
⑭
PTA, PTB
WF19810S

Figure 4. Bus Timing

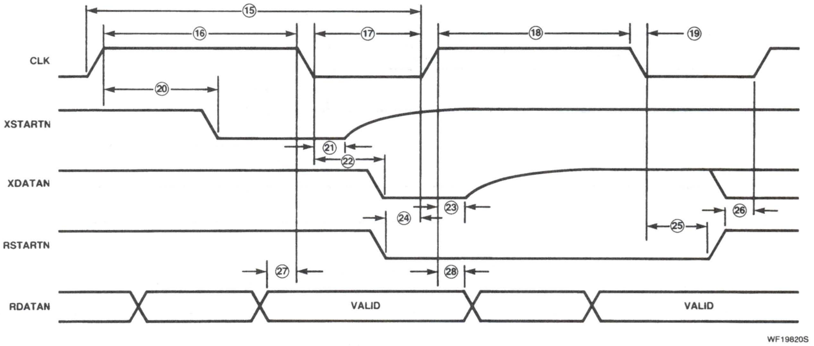

⑮
⑯
⑰
⑱
⑲
CLK
⑳
XSTARTN
㉑
㉒
XDATAN
㉓
㉔
㉖
㉕
RSTARTN
㉗
㉘
RDATAN
VALID
VALID
WF19820S

Figure 5. Serial Interface Timing

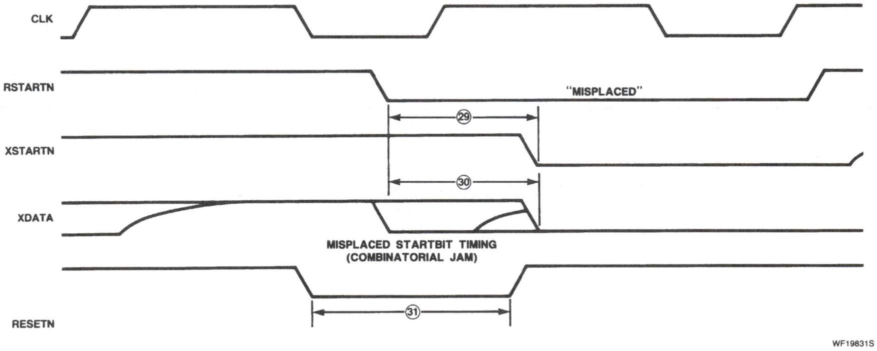

CLK
RSTARTN
"MISPLACED"
㉙
XSTARTN
㉚
XDATA
MISPLACED STARTBIT TIMING
(COMBINATORIAL JAM)
RESETN
㉛
WF19831S

Figure 6. Reset Timing

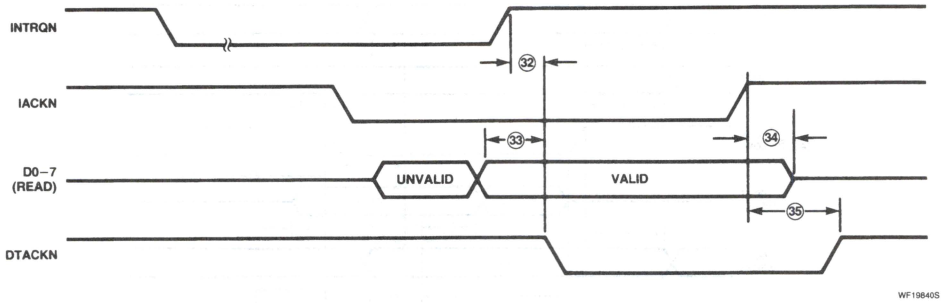

INTRQN
IACKN
D0–7 (READ)
DTACKN
UNVALID
VALID
㉜
㉝
㉞
㉟
WF19840S

Figure 7. Interrupt Timing

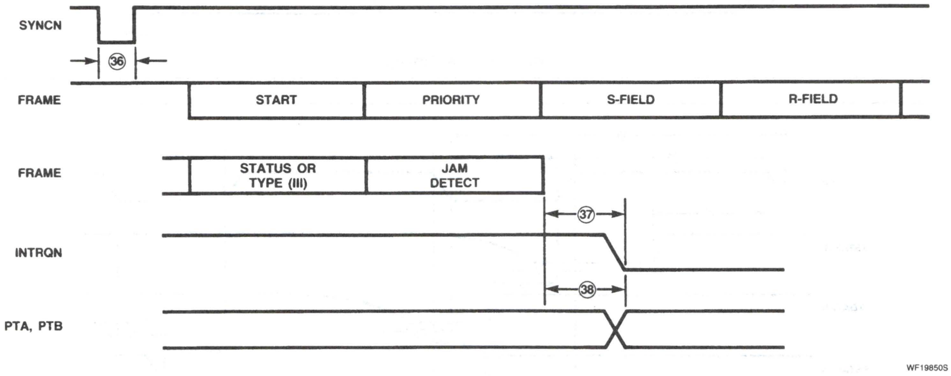

SYNCN
FRAME
FRAME
INTRQN
PTA, PTB
START
PRIORITY
S-FIELD
R-FIELD
STATUS OR TYPE (III)
JAM DETECT
㊱
㊲
㊳
WF19850S

Figure 8. Frame Timing

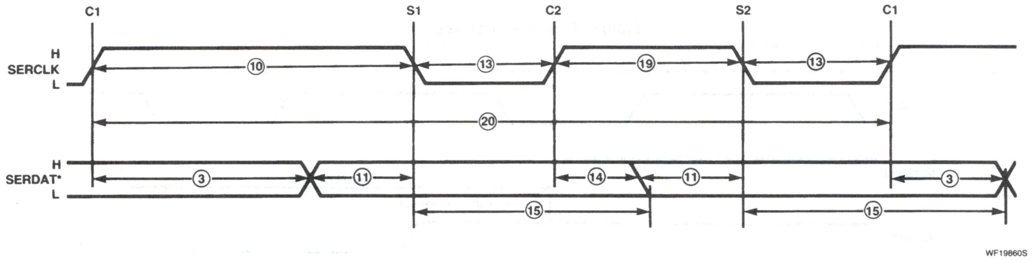

C1
S1
C2
S2
C1
H
SERCLK
L
H
SERDAT*
L
⑩
⑬
⑲
⑬
⑳
③
⑪
⑭
⑪
③
⑮
⑮
WF19860S

Figure 9. Backplane Medium Waveforms and Timing

Timing RULES for Single-Subrack CLOCK SOURCE+BRIDGE Group (Figure 9)

DESCRIPTION	CONDITIONS	MIN	NOM	MAX
10. Rule 7.5:				
A CLOCK SOURCE + BRIDGE group MUST NOT drive SERCLK low for its S1 transition until this minimum time after the C1 transition, and it MUST do so within this maximum time after C1.		175	187.5	200
13. Rule 7.6:				
A CLOCK SOURCE + BRIDGE group MUST drive SERCLK low for at least this minimum time, and it MUST NOT drive SERCLK low for more than this maximum time.		25	31.25	45
19. Rule 7.7:				
A CLOCK SOURCE + BRIDGE group MUST NOT drive SERCLK low for its S2 transition until this minimum time after the C2 transition, and it MUST do so within this maximum time after C2.		82	93.75	100
20. Rule 7.8:				
In addition to meeting the above RULES, a CLOCK SOURCE + BRIDGE module MUST drive SERCLK so that the time between successive C1 transitions is between these minimum and maximum values.		340	343.75	347
3. Rule 7.9:				
A BUS ACCESS module MUST not change SERDAT* from the value established in a bit, until SERCLK goes high in its C1 transition at the start of the next bit. RULE 7.10: IF a BUS ACCESS module is signalled by its Link Layer modules to send a zero-bit followed by a one-bit, THEN it MUST drive SERDAT* low within this maximum time after the C1 transition between these two bits. RULE 7.11: IF a BUS ACCESS module is signalled by its Link Layer modules to send a one-bit or a start bit followed by a zero-bit, THEN it must release SERDAT* within this maximum time after the C1 transition between these two bits.		0		 25 25
10. Observation 7.1:				
A BUS ACCESS module is guaranteed this minimum time between the C1 and S1 transitions of SERCLK.	Subrack /4 Rate Slower	165 650 10T + 25		
11. Observation 7.2:				
A BUS ACCESS module is guaranteed that SERDAT* is valid for this minimum time before SERCLK goes low.		10		
13. Observation 7.3:				
A BUS ACCESS module is guaranteed this minimum low time on SERCLK		15		
14. Rule 7.12:				
IF a BUS ACCESS module is signalled by its Link Layer modules to send a start-bit, AND it samples SERDAT* high at the S1 transition of SERCLK, THEN it MUST NOT drive SERDAT* low until after the C2 transition of SERCLK, and it MUST do so within this maximum time after C2.		0		25
15. Observation 7.4:				
A BUS ACCESS module is guaranteed that SERDAT* will not change until at least this long after SERCLK goes low.		5		
19. Observation 7.5:				
A BUS ACCESS module is guaranteed these minimum and maximum times between the C2 and S2 transitions of SERCLK.	Subrack /4 Rate Slower	72 275 4T + 25		110 475 8T – 25

Signetics

SCB68430
Direct Memory Access Interface (DMAI)

Product Specification

Microprocessor Products

DESCRIPTION

The SCB68430 Direct Memory Access Interface (DMAI) is a single channel interface circuit which is intended to complement the performance and architectural capabilities of the SCN68000 microprocessor. The DMAI functions by transferring a series of operands (data) between memory and a device: operand sizes may be byte, word, or long word. A block is a sequence of operands: the number of operands in the block is determined by a transfer count stored within the DMAI. The SCB68430 can be programmed to utilize single cycle (cycle stealing) or burst data transfers.

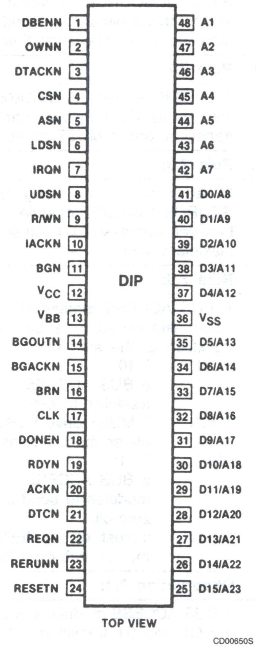
The DMAI provides two interfaces. The microprocessor interface is fully compatible with the SCN68000 microprocessor. The device interface includes lines for requesting, acknowledging, controlling, and timing the data transfers. The DMAI is a single-channel subset of the other 68000 family DMA controllers (68440 and 68450). It is software compatible with these devices and provides similar interfacing signals to both the system bus and the device.

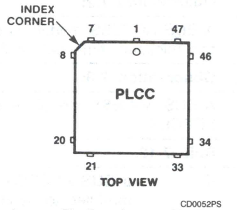
The SCB68430 is constructed using Signetics ISL bipolar technology.

FEATURES

- **Bus compatible with SCN68000 microprocessor**
- **Software compatible with other 68K family DMA controllers**
- **Single address transfers**
- **Cycle steal and burst mode operation**
- **Bus arbitration daisy chain**
- **Automatic rerun on bus error**
- **Supports 32-bit transfers for VME bus**
- **Supports SCN68000 vectored interrupts**
- **24-bit address counter**
- **16-bit transfer counter**
- **Maximum transfer rate of 5 Mbytes per sec**
- **Signetics ISL bipolar technology**

PIN CONFIGURATIONS


DBENN 1
OWNN 2
DTACKN 3
CSN 4
ASN 5
LDSN 6
IRQN 7
UDSN 8
R/WN 9
IACKN 10
BGN 11
VCC 12
VBB 13
BGOUTN 14
BGACKN 15
BRN 16
CLK 17
DONEN 18
RDYN 19
ACKN 20
DTCN 21
REQN 22
RERUNN 23
RESETN 24
48 A1
47 A2
46 A3
45 A4
44 A5
43 A6
42 A7
41 D0/A8
40 D1/A9
39 D2/A10
38 D3/A11
37 D4/A12
36 VSS
35 D5/A13
34 D6/A14
33 D7/A15
32 D8/A16
31 D9/A17
30 D10/A18
29 D11/A19
28 D12/A20
27 D13/A21
26 D14/A22
25 D15/A23
DIP
TOP VIEW
CD00650S

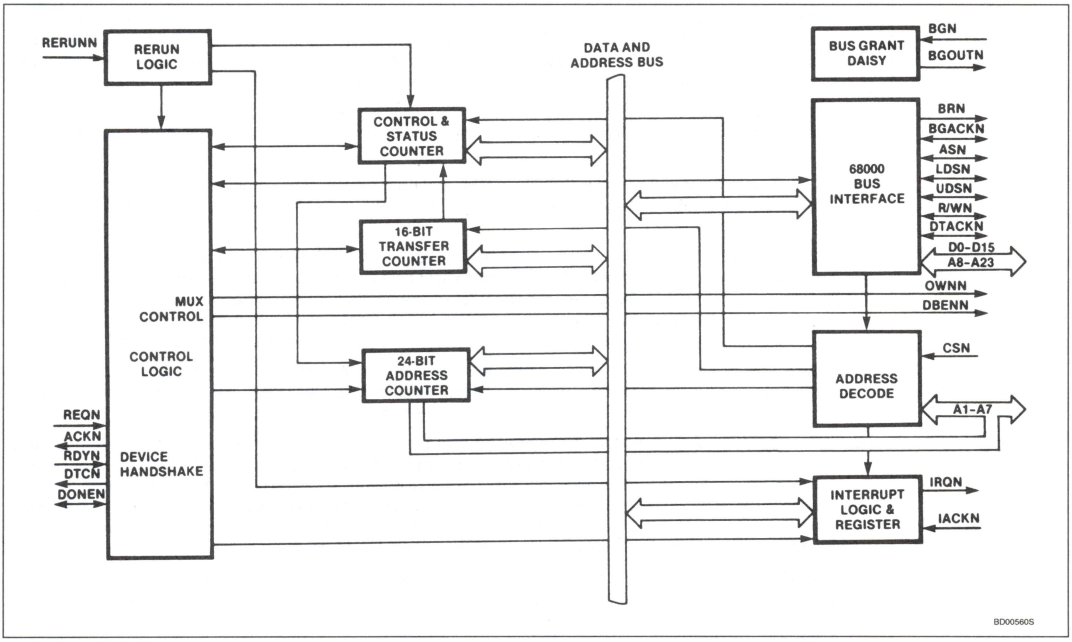

INDEX CORNER
7
1
47
8
46
PLCC
20
34
21
33
TOP VIEW
CD0052PS

Pin	Function	Pin	Function	Pin	Function
1	NC	19	CLK	36	D7/A15
2	DBENN	20	DONEN	37	D6/A14
3	OWNN	21	RDYN	38	D5/A13
4	DTACKN	22	ACKN	39	V_{SS}
5	CSN	23	DTCN	40	NC
6	ASN	24	REQN	41	D4/A12
7	LDSN	25	RERUNN	42	D3/A11
8	IRQN	26	RESETN	43	D2/A10
9	UDSN	27	NC	44	D1/A9
10	R/WN	28	D15/A23	45	D0/A8
11	IACKN	29	D14/A22	46	A7
12	BGN	30	D13/A21	47	A6
13	V_{CC}	31	D12/A20	48	A5
14	NC	32	D11/A19	49	A4
15	V_{BB}	33	D10/A18	50	A3
16	BGOUTN	34	D9/A17	51	A2
17	BGACKN	35	D8/A16	52	A1
18	BRN				

ORDERING INFORMATION

PACKAGES	$V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to $70°C$	
	10MHz	**12.5MHz**
Ceramic DIP	SCB68430CAI48	SCB68430CCI48
Plastic DIP	SCB68430CAN48	SCB68430CCN48
Plastic LCC	SCB68430CAA52	SCB68430CCA52

BLOCK DIAGRAM


RERUNN
RERUN LOGIC
DATA AND ADDRESS BUS
BUS GRANT DAISY
BGN
BGOUTN
CONTROL & STATUS COUNTER
68000 BUS INTERFACE
BRN
BGACKN
ASN
LDSN
UDSN
R/WN
DTACKN
D0-D15
A8-A23
16-BIT TRANSFER COUNTER
MUX CONTROL
OWNN
DBENN
CONTROL LOGIC
24-BIT ADDRESS COUNTER
ADDRESS DECODE
CSN
A1-A7
REQN
ACKN
RDYN
DTCN
DONEN
DEVICE HANDSHAKE
INTERRUPT LOGIC & REGISTER
IRQN
IACKN
BD00560S

Direct Memory Access Interface (DMAI) SCB68430

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	DESCRIPTION
A1 – A7	48 – 42	I/O	**Address Lines:** Active high, 3-State. In the MPU mode, these low order address lines specify which internal register of the DMAI is being accessed. In DMA mode, A1 – A7 are ouputs which provide the low order address bits of the location being accessed; 3-Stated in IDLE Mode.
A8 – A23/ D0 – D15	41 – 37 35 – 25	I/O	**Address/Data Lines:** Active high, 3-State. These lines are time multiplexed for data and address leads. The lines OWNN, RWN, CSN, and DBENN are used to control the demultiplexing of the address and data using external circuitry. In MPU mode, the bidirectional data lines (D0 – D15) are used to transfer data between the MPU and the DMAI. In the DMA mode, A8 – A23 provide the high order address bits of the location being accessed; 3-Stated in IDLE mode.
ASN	5	I/O	**Address Strobe:** Active low, 3-State. In MPU and IDLE modes, ASN is an input which indicates that the current bus master has placed a valid address on the bus. It is monitored by the DMAI during bus arbitration to ascertain that the previous bus master has completed the current bus cycle. In DMA mode, it is an output indicating that the DMAI has placed a valid address on the bus.
UDSN	8	I/O	**Upper Data Stobe:** Active low, 3-State. In MPU and IDLE modes, UDSN is an input which indicates that the upper data byte of the addressed word is being addressed. In DMA mode, it is an output with the same meaning.
LDSN	6	I/O	**Lower Data Strobe:** Active low, 3-State. In MPU and IDLE modes, LDSN is an input which indicates that the lower data byte of the addressed word is being addressed. In DMA mode, it is an output with the same meaning.
R/WN	9	I/O	**Read/Write:** Active high for read, low for write, 3-State. In MPU mode, R/WN is an input which controls the direction of data flow through the DMAI's input/output data bus interface and, if required, through an external data bus buffer. R/WN high causes the DMAI to place the data from the addressed register on the data bus, while R/WN low causes the DMAI to accept data from the data bus. In DMA mode, R/WN is an output to memory and I/O controllers indicating the type of bus cycle. It is held 3-Stated during IDLE mode.
CSN	4	I	**Chip Select:** Active low. When low, places the DMAI into the MPU mode. This input signal is used to select the DMAI for programmed data transfers. These transfers take place over D0 – D15 as controlled by the R/WN and A1 – A7 inputs. The DMAI is deselected when CSN is high. CSN is ignored during DMA mode.
DTACKN	3	I/O	**Data Transfer Acknowledge:** Active low, 3-State. In MPU mode, DTACKN is asserted on a write cycle to indicate that the data on the bus has been latched, and on a read cycle or interrupt acknowledge cycle to indicate that valid data is present on the bus. The signal is negated (driven high) when completion of the cycle is indicated by negation of the CSN or IACKN input, and returns to the inactive third state a short time after it is negated. In DMA Mode, DTACKN is an input monitored by the DMAI to determine when the addressed device (memory) has latched the data (write cycle) or put valid data on the bus (read cycle).
RESETN	24	I	**Master Reset:** Active low. Assertion of this pin clears internal control registers (See Table 1), initializes the interrupt vector register to H'0F', and sets the status register to the default value B'0000 000X', where X is the state of RDYN. All bidirectional I/O lines are 3-Stated and the DMAI is placed in the IDLE mode.
CLK	17	I	**Clock:** Active high. Usually the system clock, but may be any clock meeting the electrical specifications. Used by the DMAI to synchronize device functions and external control lines, and may not be gated off at any time.
IRQN	7	O	**Interrupt Request:** Active low, open collector. This output is asserted, if interrupts are enabled, upon end of transfer, on occurrence of a bus error, and on receipt of an abort from the MPU. The CPU can read the status register to determine the interrupting condition(s), or can respond with an interrupt acknowledge cycle to cause the DMAI to output an interrupt vector on the data bus.
IACKN	10	I	**Interrupt Acknowledge:** Active low. When asserted, indicates that the current cycle is an interrupt acknowledge cycle. The DMAI normally responds by placing the contents of the interrupt vector register on the data bus and asserting DTACKN. IACKN is not serviced if the DMAI has not generated an interrupt request.
BRN	16	O	**Bus Request:** Active low, open collector. BRN is asserted by the DMAI to request ownership of the bus after a DMA request is sensed on the REQN input from the I/O device. It is negated when the bus has been granted (BGN low) and BGACKN has been asserted, or, in burst DMA request mode, if the I/O device negates its request at least one clock cycle before BGACKN is asserted.
BGN	11	I	**Bus Grant:** Active low. BGN indicates to the DMAI that it is to be the next bus master. This signal is originated by the MPU and propagated via a daisy chain or other prioritization mechanism. After BGN is asserted, the DMAI waits until DTACKN, ASN, and BGACKN have become inactive before assuming ownership of the bus by asserting BGACKN.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.	TYPE	DESCRIPTION
BGOUTN	14	O	**Bus Grant Output:** Active low. Daisy chain output which is asserted by the DMAI when BGN is asserted and the DMAI does not have a bus request pending.
BGACKN	15	I/O	**Bus Grant Acknowledge:** Active low, 3-State. As an input, BGACKN is monitored by the DMAI during the bus arbitration cycle to determine when it can assume ownership of the bus (BGACKN negated). In DMA mode, it is asserted by the DMAI to indicate that it is the bus master. 3-Stated in MPU and IDLE modes.
RERUNN	23	I	**Rerun:** Active low. This input is asserted by external error detect logic to indicate a bus error. In DMA mode, the DMAI stops operation and 3-States the data, address, and control lines, except BGACKN. It remains halted until RERUNN becomes inactive, and then re-tries the last bus cycle. If RERUNN is asserted again, the DMAI sets the ERR bit in the status register, stops DMA operation, releases the bus, and interrupts the CPU, if interrupts are enabled, responding with a special interrupt vector when IACKN is asserted. Not monitored in MPU and IDLE modes.
REQN	22	I	**DMA Request:** Active low. This input from the I/O device requests service from the DMAI and causes the DMAI to request control of the bus. In burst mode, the input is level sensitive, and the DMAI releases the bus after REQN is negated and the current DMA cycle is completed. In cycle steal mode, the REQN input is negative edge triggered. A negative going edge must occur at least one clock cycle before DTCN is asserted to accomplish continuous transfer cycles but not earlier than beginning of master cycle.
ACKN	20	O	**DMA Request Acknowledge:** Active low. ACKN is asserted by the DMAI to indicate that it has gained the bus and the requested bus cycle is now beginning. It is asserted at the beginning of every bus cycle after ASN has been asserted, and is negated at the end of every bus cycle.
RDYN	19	I	**Device Ready:** Active low. RDYN is asserted by the requesting device to indicate to the DMAI that valid data has either been stored or put on the bus. If negated, it indicates that the data has not been stored or presented, causing the DMAI to enter wait states. RDYN can be held low continuously if the device is fast enough so that wait states are not required.
DTCN	21	O	**Device Transfer Complete:** Active low. In DMA mode, DTCN is asserted by the DMAI to indicate to the device that the requested data transfer is complete. On a write to memory operation, it indicates that the data provided by the device has been successfully stored. On a read from memory operation, it indicates to the device that the data from memory is present on the data bus and should be latched.
DONEN	18	I/O	**Done:** Active low, open collector. As an output, DONEN is asserted by the DMAI concurrent with the ACKN output to indicate to the device that the transfer count is exhausted and that the DMAI's operation is completed as a result of that transfer. As an input, if asserted by the device before the transfer count became zero, it causes the DMAI to abort service and generate an interrupt request, if interrupts are enabled.
OWNN	2	O	**Own:** Active low, open collector. This output is asserted by DMAI during the DMA mode to indicate bus mastership. It can be used to enable external address/data and control buffers. Inactive in MPU and IDLE modes.
DBENN	1	O	**Data Bus Enable:** Active low, open collector. Asserted by the DMAI when CSN is asserted or when IACKN is asserted and the DMAI has an interrupt request pending. Can be used to enable bidirectional data buffers for D0 – D15. Inactive in IDLE and DMA mode.
V_{CC}	12	I	**Power Supply:** +5V power input.
V_{BB}	13	I	**Power Supply:** +1.5V power input.
V_{SS}	36	I	**Ground:** Signal and power ground input.

PIN DESCRIPTION

The Pin Description table describes the function of each of the pins of the DMAI. Signal names ending in 'N' are active low. All other signals are active high. In the descriptions, 'MPU mode' refers to the state when the DMAI is chip selected. The term 'DMA mode' refers to the state when the DMAI assumes ownership of the bus. The DMAI is in the 'IDLE mode' at all other times.

In this data sheet signals are discussed using the terms 'active' and 'inactive' or 'asserted' and 'negated' independent of whether the signal is active in the high (logic one) state or the low (logic zero) state. Refer to the individual pin descriptions for the definition of the active level of each signal.

REGISTERS AND COUNTERS

Register Map

The internal accessible register organization of the DMAI is shown in Table 1. The following rules apply to all registers:

1. A read from a reserved location in the map results in a read from the 'null register'. The null register returns all ones for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle but no write occurs.
2. Unused bits of a defined register are read as indicated in the register descriptions.
3. All registers are addressable as 8-bit quantities. To facilitate operation with the 68K MOVEP instruction, addresses are ordered such that certain sets of registers may also be accessed as words or long words.

The operation of the DMAI is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The contents of certain control and status registers are initialized on RESET.

To provide compatibility with the other 68K family DMA controllers, control and status bits are mapped in bit positions equivalent to where they are located in the register map of the other devices. Bits which are used in the other devices but not in the DMAI are assigned default values. If upward compatibility to the other controllers is required, the programmer should use these default values when writing the control words to the registers, although they have no effect in the DMAI. When a register is read, the default value is returned regardless of the value used when the register is programmed. The default value is indicated by '(x)' in unused bit positions in the register formats, which are illustrated in Table 2.

Table 1. DMAI Address Map

ADDRESS BITS[1, 2] 7 6 5 4 3 2 1 0	ACRONYM	REGISTER NAME	MODE	AFFECTED BY RESET
d d 0 0 0 0 0 0	CSR	Channel Status Register	R/W[3]	Yes
d d 0 0 0 0 0 1	CER	Channel Error Register	R	Yes
d d 0 0 0 0 1 0		Reserved		
d d 0 0 0 0 1 1		Reserved		
d d 0 0 0 1 0 0	DCR	Device Control Register	R/W	Yes
d d 0 0 0 1 0 1	OCR	Operation Control Register	R/W	Yes
d d 0 0 0 1 1 0	SCR	Sequence Control Register	R/W[4]	No
d d 0 0 0 1 1 1	CCR	Channel Control Register	R/W	Yes
d d 0 0 1 0 0 0		Reserved		
d d 0 0 1 0 0 1		Reserved		
d d 0 0 1 0 1 0	MTCH	Memory Transfer Counter High	R/W	No
d d 0 0 1 0 1 1	MTCL	Memory Transfer Counter Low	R/W	No
d d 0 0 1 1 0 0	MACH	Memory Address Counter High	R/W[4]	No
d d 0 0 1 1 0 1	MACMH	Memory Address Counter Middle High	R/W	No
d d 0 0 1 1 1 0	MACML	Memory Address Counter Middle Low	R/W	No
d d 0 0 1 1 1 1	MACL	Memory Address Counter Low	R/W	No
d d 0 1 d d d d		Reserved		
d d 1 0 0 0 d d		Reserved		
d d 1 0 0 1 0 0		Reserved		
d d 1 0 0 1 0 1	IVR	Interrupt Vector Register – Normal	R/W	Yes
d d 1 0 0 1 1 0		Reserved		
d d 1 0 0 1 1 1	IVR	Interrupt Vector Register – Error	R/W	Yes
d d 1 0 1 0 d d		Reserved		
d d 1 0 1 1 0 0		Reserved		
d d 1 0 1 1 0 1	CPR	Channel Priority Register	R/W[4]	No
d d 1 0 1 1 1 0		Reserved		
d d 1 0 1 1 1 1		Reserved		
d d 1 1 d d d d		Reserved		

NOTES:

1. A0 = 0 for UDSN asserted, A0 = 1 for LDSN asserted.
2. 'd' designates don't care.
3. A write to this register may perform a status resetting operation.
4. This register is a dummy register present only to provide compatibility with other 68K family DMA controllers. A write to this register has no effect on the DMAI.

Table 2. Register Bit Formats

DEVICE CONTROL REGISTER

	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
DCR	**EXTERNAL REQUEST MODE** **0 = BURST** **1 = CYCLE STEAL**	**NOT USED (0)**	**NOT USED (1)**	**NOT USED (1)**	**NOT USED (*)**	**NOT USED (0)**	**NOT USED (0)**	**NOT USED (0)**

NOTE:

*Should be programmed as '0' for SIZE (OCR[5:4]) = 00 and as '1' otherwise. When read, the value of this bit is OCR[5], .OR.OCR[4].

Table 2. Register Bit Formats (Continued)

OPERATION CONTROL REGISTER (OCR)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
OCR	DIRECTION	NOT USED (0)	OPERAND SIZE		NOT USED (0)	NOT USED (0)	NOT USED (1)	NOT USED (0)
	0 = MEM TO DEV 1 = DEV TO MEM		00 = BYTE 01 = WORD (16 BIT) 10 = LONG WORD* 11 = WORD (32-BIT)*					

NOTE:
*Long word and 32-bit word modes are not supported by 68440. 32-bit word mode is not supported by 68450.

SEQUENCE CONTROL REGISTER (SCR)

	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
SCR	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (1)	NOT USED (0)	NOT USED (0)

CHANNEL CONTROL REGISTER (CCR)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
CCR	START	NOT USED (0)	NOT USED (0)	SOFTWARE ABORT	INTERRUPT ENABLE	NOT USED (0)	NOT USED (0)	NOT USED (0)
	0 = NO 1 = YES			0 = NO 1 = YES	0 = NO 1 = YES			

CHANNEL STATUS REGISTER (CSR)

	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
CSR	CHANNEL OPERATION COMPLETE	NOT USED (0)	NORMAL DEVICE TERMINATE	ERROR	CHANNEL ACTIVE	NOT USED (0)	NOT USED (0)	READY INPUT STATE
	0 = NO 1 = YES		0 = NO 1 = YES	0 = NO 1 = YES	0 = NO 1 = YES			0 = LOW 1 = HIGH

CHANNEL ERROR REGISTER (CER)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
CER	NOT USED (0)	NOT USED (0)	NOT USED (0)	ERROR CODE				
				00000 = NO ERROR 01001 = BUS ERROR 10001 = SOFTWARE ABORT				

CHANNEL PRIORITY REGISTER (CPR)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
CPR	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)

Device Control Register (DCR)

[15] External Request Mode

This bit selects whether the DMAI operates in burst or cycle steal mode.

0 Burst mode. This mode allows a device to request the transfer of multiple operands using consecutive bus cycles. In this mode the request (REQN) line is an active low input which is asserted by the device to request an operand transfer. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. If the request line is active when the DMAI asserts ACKN, and remains active at least until the DMAI asserts device transfer complete (DTCN), the DMAI recognizes a valid request for another operand, which will be transferred during the next bus cycle. If the request line is negated before the DMAI asserts DTCN, the DMAI relinquishes the bus and waits for the

next request, but the current transfer will be completed.

1 Cycle steal mode. In this mode, the device requests an operand transfer by generating a falling edge on the request (REQN) line. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. The request line must be in the inactive state for at least one clock cycle before a request is made. After a request has been asserted, it must remain at the assertion level for at least one clock cycle. If another request is received before the first operand part of a former request is acknowledged, the second request is not recognized. Normally, the DMAI will relinquish the bus after servicing a valid request. However, if the device generates a new request before the DMAI asserts DTCN for the last operand part, the DMAI will retain ownership of the bus and that request will be serviced before the bus is relinquished.

Operation Control Register (OCR)

[7] Direction

0 Transfer is from memory to device.

1 Transfer is from device to memory.

[5:4] Operand Size

The programming of these bits determine whether UDSN, LDSN, or both are generated during the transfer cycle and the increment by which the memory address counter (MAC) is changed in each transfer cycle.

00 Byte. The operand size is 8 bits. The MAC is incremented by one after each operand transfer. If the LSB of the MAC is a '0', UDSN is asserted during the transfer. If the LSB of a MAC is a '1', LDSN is asserted during the transfer. The transfer counter decrements by one before each byte is transferred.

01 Word. The operand size is 16 bits. The MAC is incremented by two after each operand transfer. The value of the LSB of the MAC is ignored and both UDSN and LDSN are asserted during the transfer. The transfer counter decrements by one before each word is transferred.

10 Long word. The operand size is 32 bits. The operand is transferred as two 16-bit words. The MAC is incremented by two after each 16-bit word is transferred. The value of the LSB of the MAC is ignored and both UDSN and LDSN are asserted during the transfer. The transfer counter decrements by one before the entire long word is transferred. Note that this mode is not implemented in the 68440.

11 Double word. The operand size is 32 bits. The operand is transferred as a single 32-bit word. The MAC is incremented by four after each operand transfer. The value of the two LSBs of the MAC is ignored (the A1 output will always be a zero in this mode) and both UDSN and LDSN are asserted during the transfer. The transfer counter decrements by one before the double word is transferred. Note that this mode is not implemented in the 68440 or 68450; it is included in the DMAI to support VME bus operations.

Sequence Control Register (SCR)

This register serves no function in the DMAI. It is included only to provide compatibility with the programming for the 68440 and 68450 DMA controllers.

Channel Control Register (CCR)

[7] Start Operation

0 No start pending.

1 Start operation. The start bit is set to initiate operation of the DMAI. The memory address counter and the memory transfer counter should have been previously initialized, and all bits of the channel status register (CSR) should have previously been reset. The DMAI initiates operation by clearing any pending requests, clearing the start bit, and setting the channel active bit in the CSR. The DMAI is then ready to receive requests for an operation. The channel cannot be started if any of the internal status bits in the CSR (CSR[15:11]) have not been cleared.
A pending start cannot be reset by a write to the register. START can be cleared only by the DMAI when it starts operation or by setting the software abort bit (CCR[4]).

[4] Software Abort

0 Do not abort.

1 Abort operation. Setting this bit terminates the current operation of the DMAI and places it in the IDLE state. The channel operation complete and error bits in the CSR are set, the channel active bit in the CSR is reset, and an ABORT ERROR condition is signaled in the CER. Setting this bit causes a pending start to be reset.

[3] Interrupt Enable

0 Interrupts not enabled.

1 Enable interrupts. An interrupt request is generated if the channel operation complete bit in the CSR is set. When the IACKN input is asserted, the DMAI returns the normal interrupt vector if the error bit in the CSR is not set, or the error interrupt vector if error is set.

Channel Status Register (CSR)

A read of this register provides the status of the DMAI. The COC, NDT, and ERR bits can be cleared by writing a '1' to the bit positions of the register which are to be cleared. Those bit positions which are written with a '0' remain unaffected.

[15] Channel Operation Complete

This bit is set following the termination, whether successful or not, of any DMAI operation and indicates that the DMA transfer has completed. This bit must be cleared to start another channel operation.

[13] Normal Device Termination

This bit is set when the device terminates the DMAI operation by asserting the DONEN line while the device was being acknowledged. This bit must be cleared to start another channel operation.

[12] Error

This bit is used to report that the DMAI's operation was terminated due to the occurrence of an error. The condition which caused the error can be determined by reading the channel error register (CER). This bit must be cleared to start another channel operation. When this bit is cleared, the CER is also cleared.

[11] Channel Active

This bit is set after the channel has been started and remains set until the channel operation terminates. It is then automatically cleared by the DMAI. The bit is unaffected by the write operations.

[8] Ready Input State

This bit reflects the state of the RDYN input at the time the CSR is read. The bit is a '0' if RDYN is low and a '1' if RDYN is high. This bit is unaffected by write or reset operations.

Channel Error Register (CER)

[4:0] Error Code

This field indicates the source of error when an error is indicated in CER[12]. The contents of this register are cleared when CER[12] is cleared.

00000 No error.

01001 Bus error. A bus error occurred during the last bus cycle generated by the DMAI. See rerun description in OPERATION section.

10001 Software abort. The channel operation was terminated by a software abort command. See CCR[4].

Channel Priority Register (CPR)

This register serves no function in the DMAI. It is included only to provide compatibility with the programming for the other 68K family DMA controllers.

Memory Address Counter (MACH, MACMH, MACML, MACL)

The 32-bit memory address counter is used to program the memory location where the first operand to be transferred is located or is to be transferred to, depending on the direction of transfer. The counter must be initialized prior to beginning the transfer of a block of data and then increments automatically depending on the operand length, as described in the Operation Control Register description.

Only the least significant 24 bits of the counter (MACMH, MACML, and MACL) are implemented in the DMAI. The most significant byte of the counter, MACH, is provided only to allow compatibility with programming of the 68440 and 68450. Writing to MACH has no effect on the DMAI operation. Reading MACH always returns H'00'.

Memory Transfer Counter (MTCH, MTCL)

The 16-bit memory transfer counter programs the number of operands to be transferred by the DMAI. The counter must be initialized prior to beginning the transfer of a block of data and then decrements once per operand transfer (regardless of operand size) until it reaches the terminal value of zero. Channel operation then terminates and the COC bit in the CSR will be asserted.

Interrupt Vector Register (IVR)

The IVR contains the value to be placed on the data bus upon receipt of an interrupt acknowledge from the MPU. Only the seven most significant bits of the programmed value are used by the DMAI. The output vector from the DMAI contains a zero in the least significant bit position if a normal termination occurred (error bit not set) and contains a one in the least significant bit position if termination was due to an error (error bit set).

The contents of this register are initialized to H'0F' by a reset. The value returned will be H'0F', regardless of the error state, until the register is programmed by the MPU.

To provide compatibility with the other 68K family DMA controllers, the IVR has two addresses (see Table 1). If program compatibility is required, the value written at the normal IVR address should have a zero as its LSB, and the value written at the error IVR address should be the same but with the LSB equal to one.

OPERATION

A DMAI operation proceeds in three principal phases. During the initialization phase, the MPU configures the channel control registers, loads the initial memory address and transfer count, and starts the channel. During the transfer phase, the DMAI accepts requests for transfers from the device, arbitrates for and acquires ownership of the bus, and provides for addressing and bus controls for the transfers. The termination phase occurs after the operation is complete, when the DMAI reports the status of the operation.

Operation Initiation

After having programmed the control registers, the memory address counter, and the memory transfer counter, the MPU sets the start bit (CCR[7]). The DMAI initiates the operation by clearing any pending requests, clearing the start bit, and setting the channel active bit in the CSR. The DMAI is then ready to receive valid requests for an operation.

The channel cannot be started if any of the internal status bits in the CSR (CSR[15:11]) have not been cleared. An error is not signaled if this condition occurs. The only indication of this state is that the start bit remains set in the CCR. A pending start cannot be reset by a write to the register. START can be cleared only by the DMAI when it starts operation or by setting the software abort bit (CCR[4]).

Device/DMAI Communication

Communication between the peripheral device and the DMAI is accommodated by five signal lines:

Request (REQN)

The device makes a request for service by asserting the request line. The DMAI can operate in either the burst request mode or the cycle stealing request mode, as programmed by the external request mode bit (DCR[15]).

The burst mode allows a device to request the transfer of multiple operands using consecutive bus cycles. In this mode the request line is an active low input. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. If the request line is active when the DMAI asserts ACKN, and remains active at least until the DMAI asserts device transfer complete (DTCN), the DMAI recognizes a valid request for another operand, which will be transferred during the next bus cycle. If the request line is negated before the DMAI asserts DTCN, the DMAI relinquishes the bus and waits for the next request. For long word transfers (2 × 16), the request must be asserted at least until the acknowledge for the second part of the operand has been asserted.

In the cycle steal mode, the device requests an operand transfer by generating a falling edge on the request line. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. The request line must be in the inactive state for at least one clock cycle before a request is made. After a request has been asserted, it must remain at the assertion level for at least one clock cycle. If another request is received before the first operand part of a former request is acknowledged, the second request is not recognized. Normally, the DMAI will relinquish the bus after servicing a valid request. However, if the device generates a new request before the DMAI asserts DTCN for the last operand part, the DMAI will retain ownership of the bus and that request will be serviced before the bus is relinquished. During burst mode, REQN must not be disasserted for less than one CLK period plus four RC time constants, where R is the value of the resistor used for the pullup on BRN and C has a typical value of 20pF.

Acknowledge (ACKN)

The DMAI asserts the acknowledge line, which implicitly addresses the device making the request, during transfers to and from the device. The line may be used to control buffering circuits between the data bus and the MPU bus.

Ready (RDYN)

Ready is an active low input which is asserted by the requesting device to indicate to the DMAI that valid data has either been stored or put on the bus. If negated, it indicates that the data has not been stored or presented, causing the DMAI to enter wait states until RDYN is asserted. RDYN can be held low continuously if the device is fast enough so that wait states are not required. The current state of the ready input is reflected in CSR[8].

Done (DONEN)

Done is a bidirectional active low signal. As an output, it is asserted and negated by the DMAI concurrent with the ACKN output of the last operand part to indicate to the device that the memory transfer count is exhausted and that the DMAI's operation is completed as a result of that transfer.

The DMAI also monitors the state of the line while acknowledging a device. If the device asserts DONEN, the DMAI will terminate operation after the transfer of the current operand. In this case the DMAI clears the channel active bit and sets the channel operation complete and normal device termination bits in the CSR. If both the DMAI and the device assert DONEN, the device termination is not recognized, but the operation does terminate.

Device Transfer Complete (DTCN)
DTCN is an active low output which is asserted by the DMAI to indicate to the device that the requested data transfer is complete. On a write to memory operation, it indicates that the data provided by the device has been successfully stored. On a read from memory operation, it indicates to the device that the data from memory is present on the data bus and should be latched. DTCN is not asserted if assertion of the RERUNN input terminates the bus cycle.

Bus Arbitration

Upon receiving a valid request for a transfer from the device, the DMAI will arbitrate for and obtain ownership of the system bus.

The DMAI indicates that it wishes to become the bus master by asserting its bus request (BRN) output. This is a wire-ORed signal that indicates to the MPU that some external device requires control of the bus. The processor is effectively at a lower priority level than external devices and will relinquish the bus after it has completed the last bus cycle it has started. The processor puts the bus up for external arbitration by asserting its bus grant (BGN) output. This signal may be routed through a daisy chain (such as provided by the DMAI) or through some other priority-encoded network. When the DMAI making the bus request receives the bus grant (indicated by its BGN input being asserted), it is to be the next bus master. It waits until address strobe (ASN), data transfer acknowledge (DTACKN) and bus grant acknowledge (BGACKN) become inactive and then assumes ownership of the bus by asserting its own BGACKN output. The DMAI then negates the BRN output and proceeds with the data transfer phase. After this phase is completed, the DMAI relinquishes bus ownership by negating the BGACKN output.

In burst DMA mode, detection of an active low request input after the DMAI operation has been started will begin the bus arbitration cycle. However, if the device negates its request at least one clock cycle before the DMAI asserts BGACKN, the DMAI will negate its bus request and will not assume ownership of the bus.

Data Transfers

The actual transfer of data between the memory and the device occurs during the data transfer phase. All transfers occur during a single cycle except in the case of long word operands, in which case two cycles are used to transfer the operand as two 16-bit words. The transfers take place using a 'single address' protocol; the DMAI addresses the memory via the bus address lines, while the device is implicitly addressed via the acknowledge output.

When a request is generated using the request method programmed in the control register, the DMAI obtains the bus and asserts acknowledge to notify the device that a transfer is to take place. The DMAI asserts all S68000 bus control signals needed for the transfer and holds them until the device responds with ready. The bus cycle then terminates normally. Ready may be tied low (asserted) if the device is fast enough.

When the transfer is from memory to the device, data is valid when DTACKN is asserted by the memory and remains valid until the data strobe(s) are negated. The assertion of DTCN from the DMAI can be used to latch the data, as the data strobes are not removed until one-half clock after the assertion of DTCN.

When the transfer is from device to memory, the data must be valid on the bus before the DMAI asserts the data strobe(s). The device indicates valid data by asserting ready. The DMAI then asserts the strobes and holds them asserted until the memory accepts the data, indicated by the assertion of DTACKN. The DMAI then asserts DTCN and negates the data strobes.

Flowcharts for these operations are shown in Figures 1 and 2. Refer to the timing section for the equivalent timing diagrams.

Operation Termination

Termination of the block transfer occurs under the conditions detailed below.

Terminal Count
As part of each transfer of an operand, the DMAI decrements the memory transfer counter. If this counter is decremented to zero, the operand is the last operand of the block. The DMAI operation is complete and it notifies the device of completion by asserting the DONEN output during the last operand transfer cycle. When the transfer has been completed, the channel active bit in the CSR is cleared and the COC bit is set, unless an error occurs.

Device Termination
The DMAI monitors the state of the DONEN line while acknowledging a device transfer request. If the device asserts DONEN, the DMAI will terminate operation after the transfer of the current operand. When the transfer has been completed, the DMAI clears the channel active bit and sets the COC and normal device termination bits in the CSR. If both the DMAI and the device assert DONEN, the device termination is not recognized, but the operation does terminate.

Software Abort
The software abort bit (CCR[4]) allows the MPU to abort the current operation of the DMAI. The COC and error bits in the CSR are set, the channel active bit in the CSR is cleared, and an abort error condition is signaled in the CER.

Rerun Error
The DMAI provides a rerun input (RERUNN) to indicate a bus exception condition. RERUNN must arrive prior to or in coincidence with DTACKN in order to be recognized, and the DMAI verifies that the line has been stable for two clock cycles before acting on it. The occurrence of a rerun during a DMAI bus cycle forces it to terminate the bus cycle in an orderly manner.

When the assertion of rerun is verified, the DMAI stops operation and 3-States the data, address, and control lines, except BGACKN, so that it retains ownership of the bus. It remains halted until rerun becomes inactive, and then re-tries the last bus cycle. If rerun is asserted again, the DMAI stops DMA operation, releases the bus, sets the error and COC bits in the CSR, clears the active bit in the CSR, and sets the error code in the CER to indicate a bus error.

While stopped due to assertion of rerun, the DMAI does not generate any bus cycles and will not honor any requests until it is removed. However, the DMAI still recognizes requests.

Error Recovery Procedure
If an error occurs during a DMA transfer such that the DMAI stops the DMA operation, information is available to the operating system for an error recovery routine.

The information available to the operating system consists of the memory address counter, the memory transfer counter, and the control, status, and error registers. The DMAI decrements the memory transfer counter before attempting a DMA operation, so the register will contain the count minus one of the attempted transfer. The memory address counter will contain the address at which the DMA operation was attempted.

Reset
The reset input (RESETN) provides a means of resetting and initializing the DMAI from an external source. If the DMAI is a bus master when reset is received, the DMAI relinquishes the bus. Reset clears the control and error registers, sets all bits of the status register except CSR[8] to zero, and initializes the interrupt vector register to H'0F'.

Interrupts
The interrupt enable bit (CCR[3]) determines whether the DMAI generates interrupt requests. When the bit is set, an interrupt request is generated if the channel operation complete bit in the CSR is set. When the

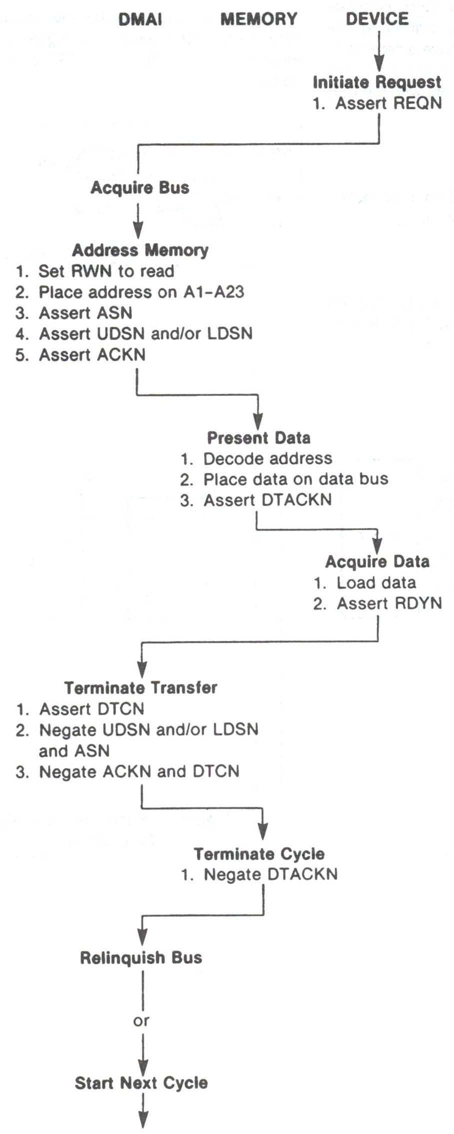

DMAI
MEMORY
DEVICE
Initiate Request
1. Assert REQN
Acquire Bus
Address Memory
1. Set RWN to read
2. Place address on A1–A23
3. Assert ASN
4. Assert UDSN and/or LDSN
5. Assert ACKN
Present Data
1. Decode address
2. Place data on data bus
3. Assert DTACKN
Acquire Data
1. Load data
2. Assert RDYN
Terminate Transfer
1. Assert DTCN
2. Negate UDSN and/or LDSN and ASN
3. Negate ACKN and DTCN
Terminate Cycle
1. Negate DTACKN
Relinquish Bus
or
Start Next Cycle
PF00460S

Figure 1. Transfer From Memory to Device Flowchart

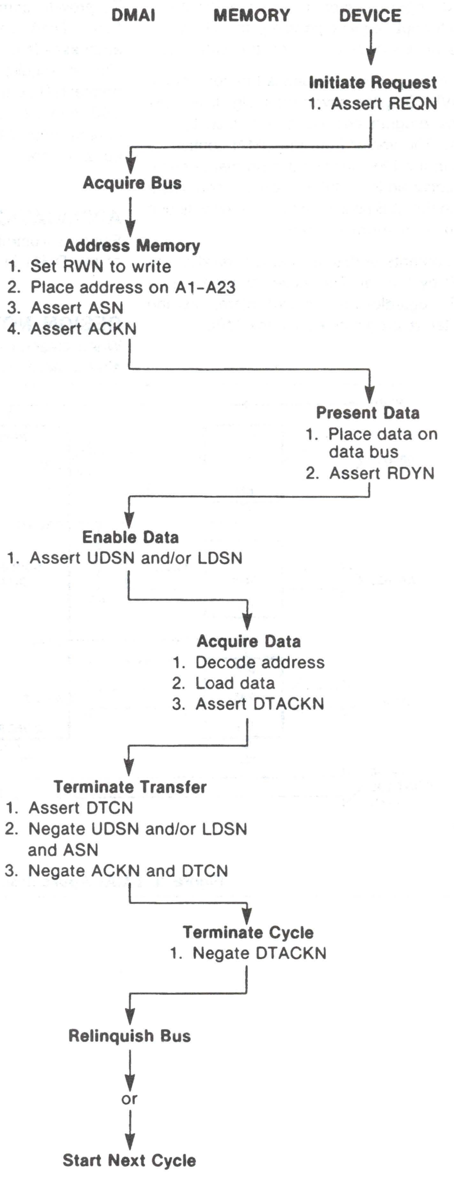

DMAI
MEMORY
DEVICE
Initiate Request
1. Assert REQN
Acquire Bus
Address Memory
1. Set RWN to write
2. Place address on A1–A23
3. Assert ASN
4. Assert ACKN
Present Data
1. Place data on data bus
2. Assert RDYN
Enable Data
1. Assert UDSN and/or LDSN
Acquire Data
1. Decode address
2. Load data
3. Assert DTACKN
Terminate Transfer
1. Assert DTCN
2. Negate UDSN and/or LDSN and ASN
3. Negate ACKN and DTCN
Terminate Cycle
1. Negate DTACKN
Relinquish Bus
or
Start Next Cycle
PF00470S

Figure 2. Transfer From Device to Memory Flowchart

IACKN input is asserted, and the DMAI has an interrupt request pending, the DMAI returns an interrupt vector on the data bus.

The interrupt vector issued is the contents of the IVR. Only the seven most significant bits of the programmed value are used by the DMAI. The vector from the DMAI contains a zero in the LSB position if a normal termination occurred (error bit not set) and contains a one in the LSB position if termination was due to an error (error bit set).

The contents of this register are initialized to H'0F' by a reset. The value returned will be H'0F', regardless of the error state, until the register is programmed by the MPU.

To provide compatibility with the other 68K family DMA controllers, the IVR has two addresses (see Table 1). If program compatibility is required, the value written at the normal IVR address should have a zero as its LSB, and the value written at the error IVR address should be the same but with the LSB equal to one.

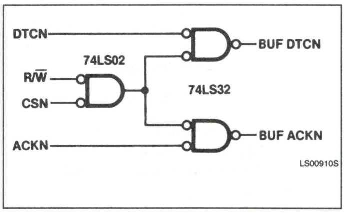
APPLICATIONS

Figure 3 illustrates a typical interconnection of the DMAI in a 68000 based system.

DESIGN NOTE

When clearing the error bit in CSR (bit 12) after a DMAI abort due to a double RERUNN, ACKN and DTCN will both go low concurrent with CSN and DTACKN for one CLK cycle.

To prevent the possibility that the device may misinterpret these signals, it is suggested that these signals be ANDed with CSN (see figure below).


DTCN
74LS02
R/W̄
CSN
ACKN
74LS32
BUF DTCN
BUF ACKN
LS00910S

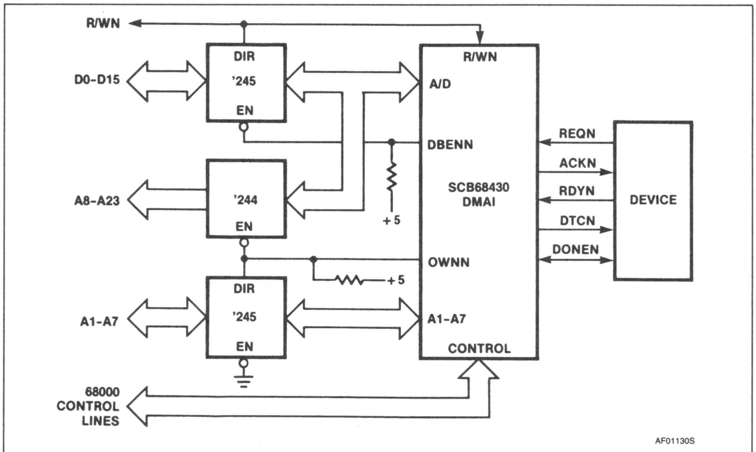

R/WN
D0-D15
DIR
'245
EN
A8-A23
'244
EN
DIR
'245
EN
A1-A7
68000
CONTROL
LINES
+5
+5
R/WN
A/D
DBENN
SCB68430
DMAI
OWNN
A1-A7
CONTROL
REQN
ACKN
RDYN
DTCN
DONEN
DEVICE
AF01130S

Figure 3. DMAI Application

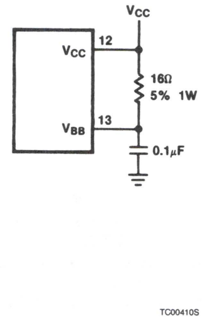

VCC
VCC
12
16Ω
5% 1W
VBB
13
0.1μF
TC00410S

Figure 4. Recommended V_{BB} Test Circuit

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}, V_{BB}	Supply voltages	−0.5 to +7.0	V
V_{IN}	Input voltage range	−0.5 to +5.5	V
T_A	Operating temperature range[2]	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, V_{BB} = Figure 4, $T_A = 0°C$ to $+70°C$[3, 4, 7]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{IL}	Input low voltage			0.8	V
V_{IH}	Input high voltage		2		V
V_{OL}	Output low voltage	$I_{OUT} = 5.3mA$		0.5	V
V_{OH}	Output high voltage, all outputs except open collector outputs[5]	$I_{OUT} = -400\mu A$	2.5		V
I_{IL}	Input low current	$V_{IN} = 0.4V$		−400	µA
I_{IH}	Input high current	$V_{IN} = 2.7V$		20	µA
I_{OC}	Open collector off state current[5]	$V_{OUT} = 2.4V$		20	µA
I_{SC}	Output short circuit current[6]	V_{CC} = Max	−40	−100	mA
I_{CC}	V_{CC} supply current	V_{CC} = Max		130	mA
I_{BB}	V_{BB} supply current			275	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other conditions other than those indicated in the Electrical Characteristics section of this data sheet is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. Parameters are valid over specified temperature range.
4. All voltage measurements are referenced to ground (V_{SS}). For testing, all signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2V as appropriate.
5. IRQN, BRN, DONEN, and OWNN are open collector outputs.
6. No more than one output should be connected to ground at one time.
7. Capacitive test load is 100pF for all pins except DTCN which has a 35pF capacitive test load.

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, V_{BB} = Figure 4, $T_A = 0°C$ to $+70°C$[3, 4, 7]

NO.	FIGURE	CHARACTERISTICS	TENTATIVE LIMITS				UNIT
			10MHz		12.5MHz		
			Min	Max	Min	Max	
1	5	A1 – A7, ASN, RWN, setup to UDSN, LDSN low	0		0		ns
2	5	D0 – D15 3-State to invalid data from ASN, CSN, and UDSN or LDSN low	10		10		ns
3	5	DTACKN 3-State to high from ASN, CSN, and UDSN or LDSN low	10		10		ns
4	5	CSN low after UDSN or LDSN low		25		25	ns
5	5, 6	DBENN low after ASN and CSN low		60		60	ns
6	5	D0 – D15 valid data from ASN, CSN, and UDSN or LDSN low		100		100	ns
7	5	DTACKN low after D0 – D15 valid data	−15	30	−15	30	ns
8	5	A1 – A7, ASN, RWN or CSN hold after UDSN and LDSN high	0		0		ns
9	5, 6	DBENN high from either ASN or CSN high		45		45	ns
10	5	D0 – D15 to 3-State from UDSN and LDSN high		80		80	ns
11	5	D0 – D15 to invalid data from UDSN and LDSN high	10		10		ns
12	5, 6	DTACKN high from UDSN and LDSN high		55		55	ns
13	5, 6	DTACK 3-State from either CSN or ASN high		85		85	ns
14	6	A1 – A7, ASN, RWN setup to UDSN, LDSN low	50		50		ns
15	6	CSN setup before UDSN or LDSN low	20		20		ns
16	6	DTACKN 3-State to high after CSN and ASN low	10		10		ns
17	6	D0 – D15 valid after UDSN or LDSN low		0		0	ns
18	6	DTACKN low from UDSN or LDSN low		100		100	ns
19	6	UDSN and LDSN low time	115		100		ns
20	6	A1 – A7 hold after UDSN and LDSN high	0		0		ns
21	6	ASN, RWN and CSN hold after UDSN and LDSN high	0		0		ns

Direct Memory Access Interface (DMAI) SCB68430

AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	FIGURE	CHARACTERISTICS	TENTATIVE LIMITS				UNIT
			10MHz		12.5MHz		
			Min	Max	Min	Max	
22	6	D0 – D15 hold after UDSN and LDSN high	0		0		ns
23	7	DBENN low from last low of ASN, IACKN, LDSN		65		65	ns
24	7	D0 – D7 valid after last low of ASN, IACKN, LDSN		105		105	ns
25	7	DTACKN 3-State to high after last low of ASN, IACKN, LDSN		100		100	ns
26	7	DTACKN low after last low of ASN, IACKN, LDSN		110		110	ns
27	7	DBENN high after first high of ASN, IACKN, LDSN		55		55	ns
28	7	D0 – D7 hold after first high of ASN, IACKN, LDSN		60		60	ns
29	7	D0 – D7 3-State after first high of ASN, IACKN, LDSN		80		80	ns
30	7	DTACKN high after first high of ASN, IACKN, LDSN		60		60	ns
31	7	DTACKN 3-State after first high of ASN, IACKN, LDSN		95		95	ns
32	8	BRN high from CLK high		65		65	ns
33	8, 11, 12	BGACKN low from CLK low		75		75	ns
34	8, 11, 12	OWNN low from CLK high		75		75	ns
35	8	BGACKN high from CLK low		75		75	ns
36	8,11,12	OWNN high from CLK high (load dependent)		50		50	ns
37	10	REQN setup before CLK low	30		30		ns
38	10	REQN hold after CLK high	20		20		ns
39	10	BRN low from CLK high		80		80	ns
41	11, 12	ASN, UDSN, LDSN, RWN 3-State to high from CLK low		75		75	ns
43	11, 12	A1 – A23 to valid ASN	0		0		ns
44	11, 12	ASN low from CLK high		65		65	ns
45	11, 12	LDSN, UDSN low from CLK high		90		90	ns
46	11, 12	ACKN low from CLK high		65		65	ns
47	11, 12	DTACKN setup to CLK high	30		30		ns
48	11, 12	RDYN setup to CLK low	30		30		ns
49	11, 12	DTCN low from CLK high		70		70	ns
50	11, 12	ASN high from CLK high		75		75	ns
51	11, 12	LDSN, UDSN, high from CLK high		90		90	ns
52	11, 12	DTACKN, RDYN hold after CLK high	0		0		ns
–	11, 12	ASN, LDSN, UDSN, high from DTCN low	–20		–20		ns
53	11, 12	ACKN high from CLK high		50		50	ns
54	11, 12	DTCN high from CLK high		50		50	ns
55	11, 12	Address valid after CLK low	10		10		ns
–	11, 12	Address valid after ASN high	0		10		ns
56	11, 12	DONEN (output) low from CLK low		120		120	ns
57	11, 12	DONEN (output) high from CLK high		50		50	ns
58	11, 12	DONEN (input) setup low before CLK low	30		30		ns
59	11, 12	DONEN (input) hold low after CLK high	0		0		ns
60	11, 12	BGACKN, ASN, UDSN, LDSN, RWN to 3-State from CLK low		75		75	ns
62	11, 12	A1 – A23 valid to 3-State from CLK high		100		100	ns
63	12	R/WN low from CLK high		65		65	ns
64	12	R/WN high from CLK high		75		75	ns
65	13	RERUNN setup low before CLK high	30		30		ns
66	13	RERUNN hold low from CLK high	20		20		ns
67	13	A1 – A23 to idle state from CLK low		100		100	ns
68	13	A1 – A23 to valid after CLK low		85		85	ns

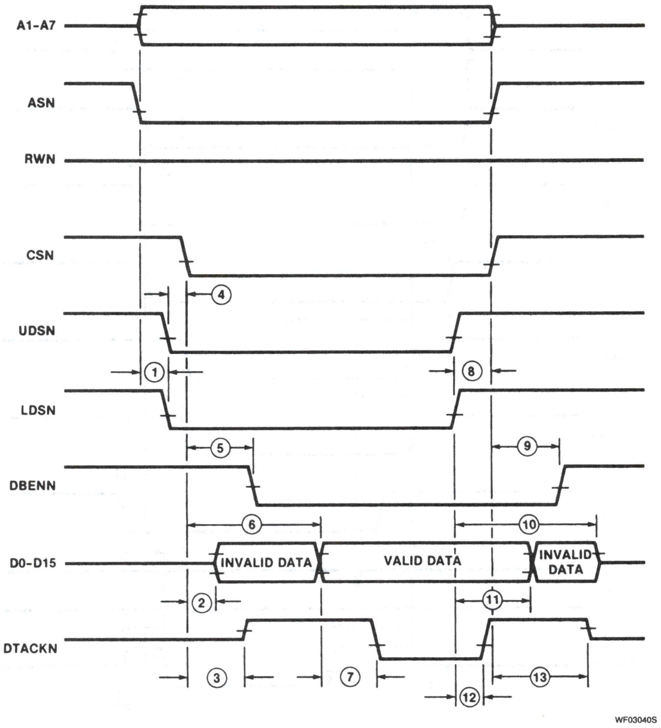

A1-A7
ASN
RWN
CSN
UDSN
LDSN
DBENN
D0-D15
DTACKN
INVALID DATA
VALID DATA
INVALID DATA
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
WF03040S

Figure 5. DMAI Read Timing

Direct Memory Access Interface (DMAI) SCB68430

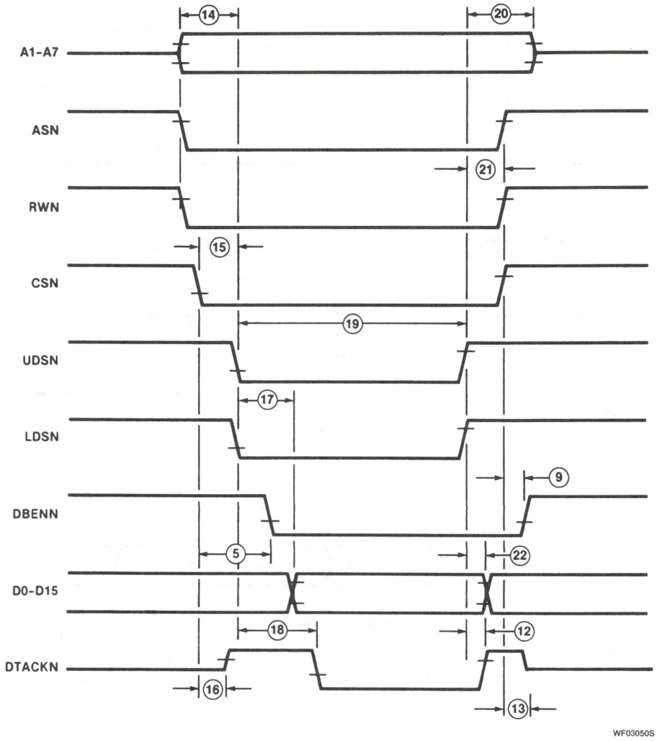

A1-A7
ASN
RWN
CSN
UDSN
LDSN
DBENN
D0-D15
DTACKN
14
20
21
15
19
17
9
5
22
18
12
16
13
WF03050S

Figure 6. DMAI Write Timing

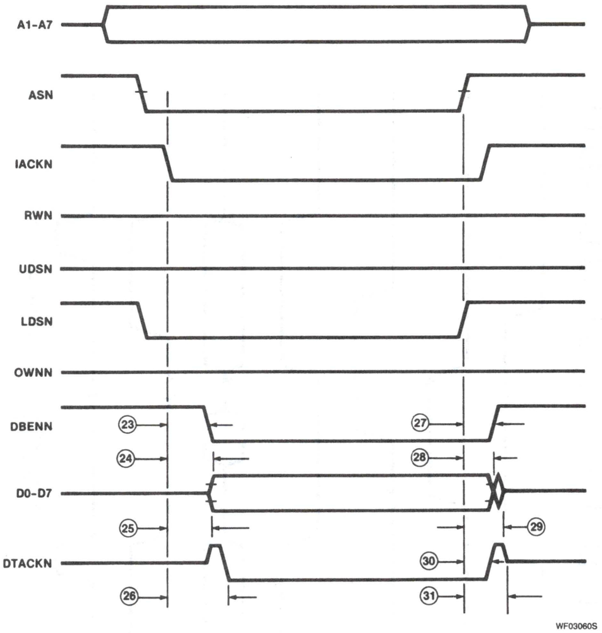
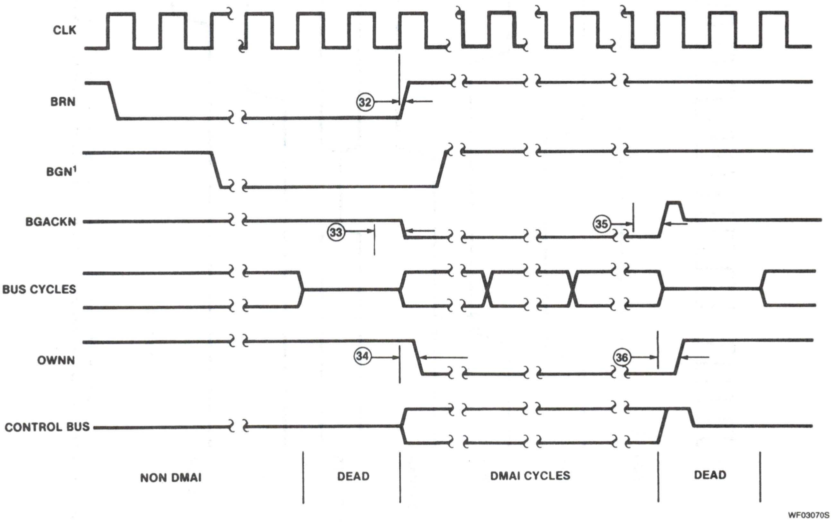
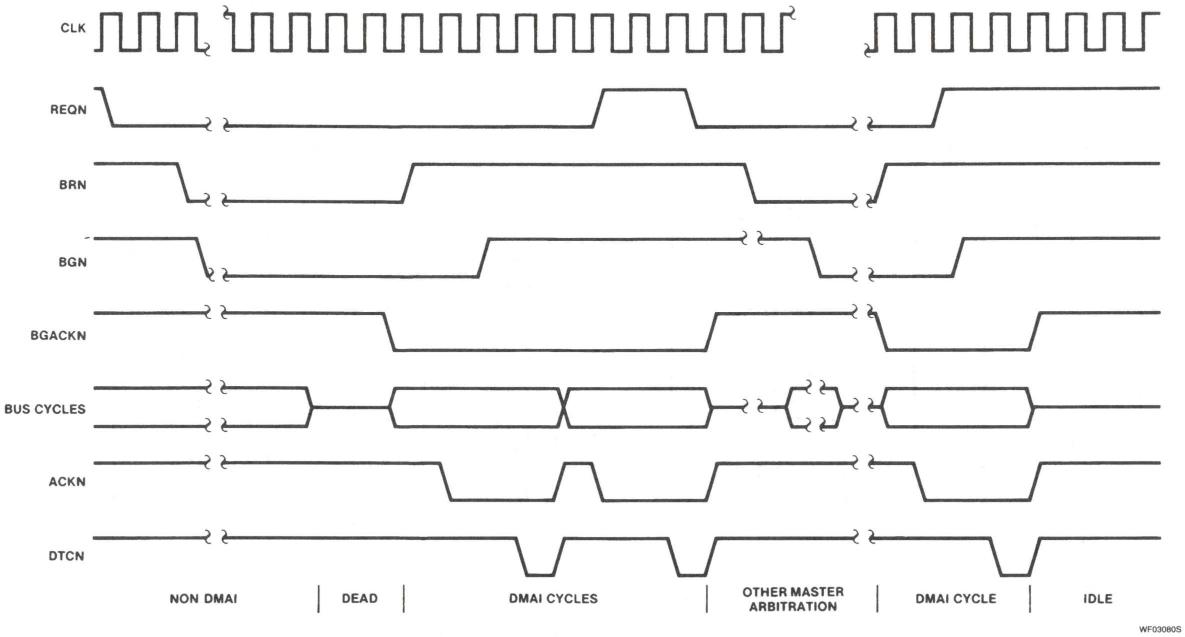

A1-A7
ASN
IACKN
RWN
UDSN
LDSN
OWNN
DBENN
D0-D7
DTACKN
㉓
㉔
㉕
㉖
㉗
㉘
㉙
㉚
㉛
WF03060S

Figure 7. CPU IACK Cycle to DMAI


CLK
BRN
BGN¹
BGACKN
BUS CYCLES
OWNN
CONTROL BUS
㉜
㉝
㉞
㉟
㊱
NON DMAI
DEAD
DMAI CYCLES
DEAD
WF03070S

NOTES:
1. Device will become master if BGN is asserted concurrent with or later than REQN (same clock edge or later).
2. ASN, DTACKN and BGACKN must be negated in order for DMAI to become master. Timing assumes all these happen concurrent with BGN — if not, it is from the latest signal which is negated.

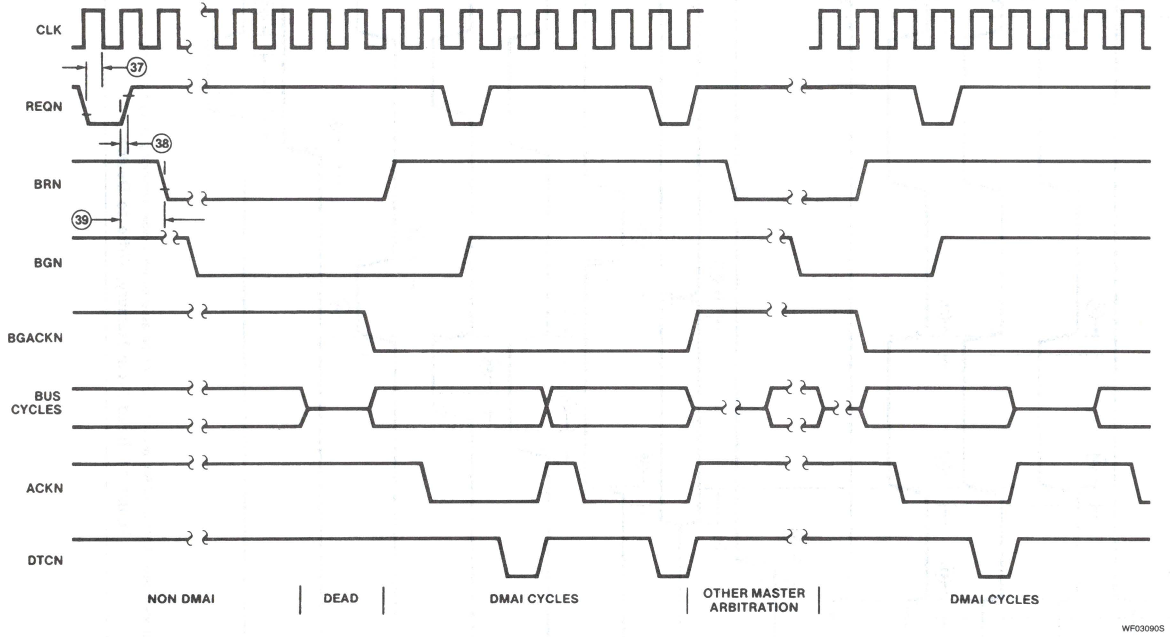
Figure 8. DMAI Bus Arbitration Timing


CLK
REQN
BRN
BGN
BGACKN
BUS CYCLES
ACKN
DTCN
NON DMAI
DEAD
DMAI CYCLES
OTHER MASTER ARBITRATION
DMAI CYCLE
IDLE
WF03080S

NOTES:

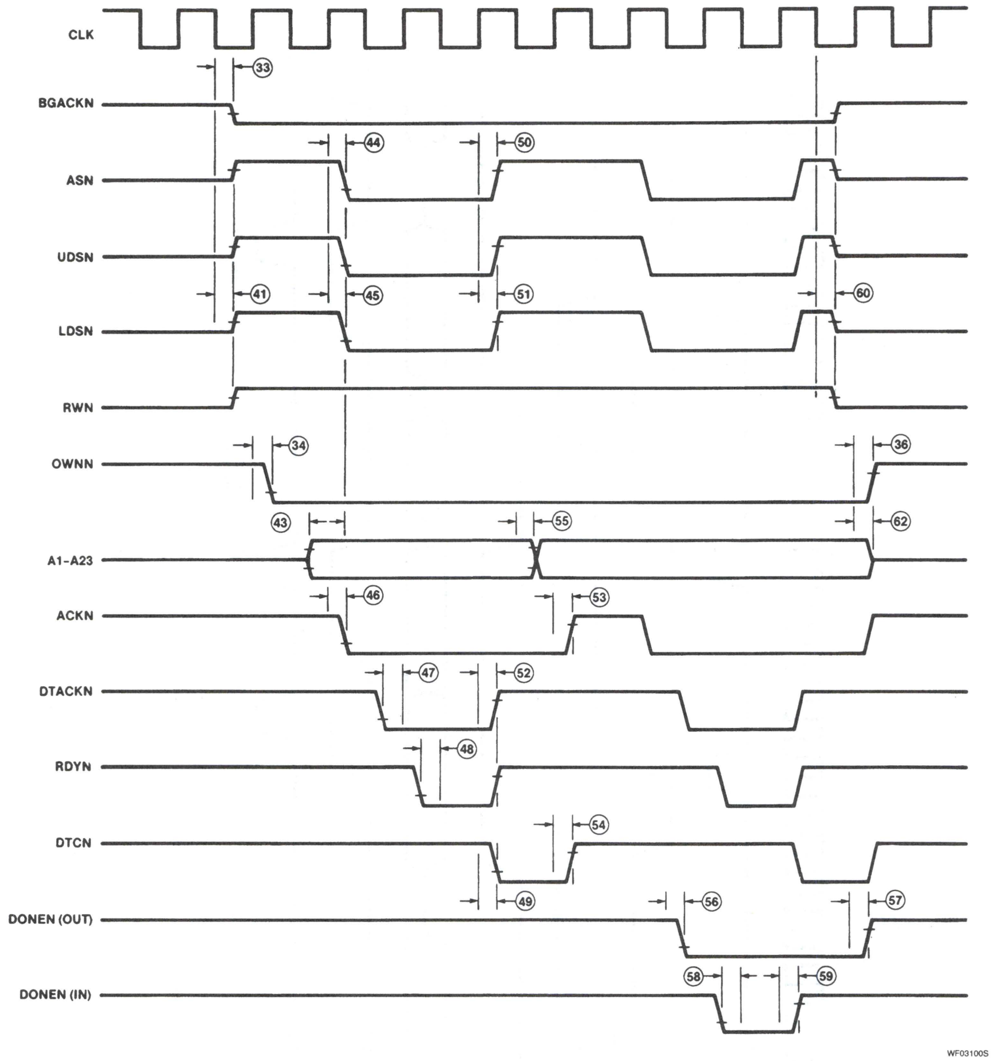
1. To maintain mastership next bus cycle, REQN must remain asserted at least until 1/2 clock cycle after DTCN is asserted.
2. To guarantee that mastership is relinquished next cycle, REQN must be negated no later than 1/2 clock cycle prior to DTCN.

Figure 9. DMAI Burst Mode Request Timing


CLK
REQN
BRN
BGN
BGACKN
BUS CYCLES
ACKN
DTCN
37
38
39
NON DMAI
DEAD
DMAI CYCLES
OTHER MASTER ARBITRATION
DMAI CYCLES
WF03090S

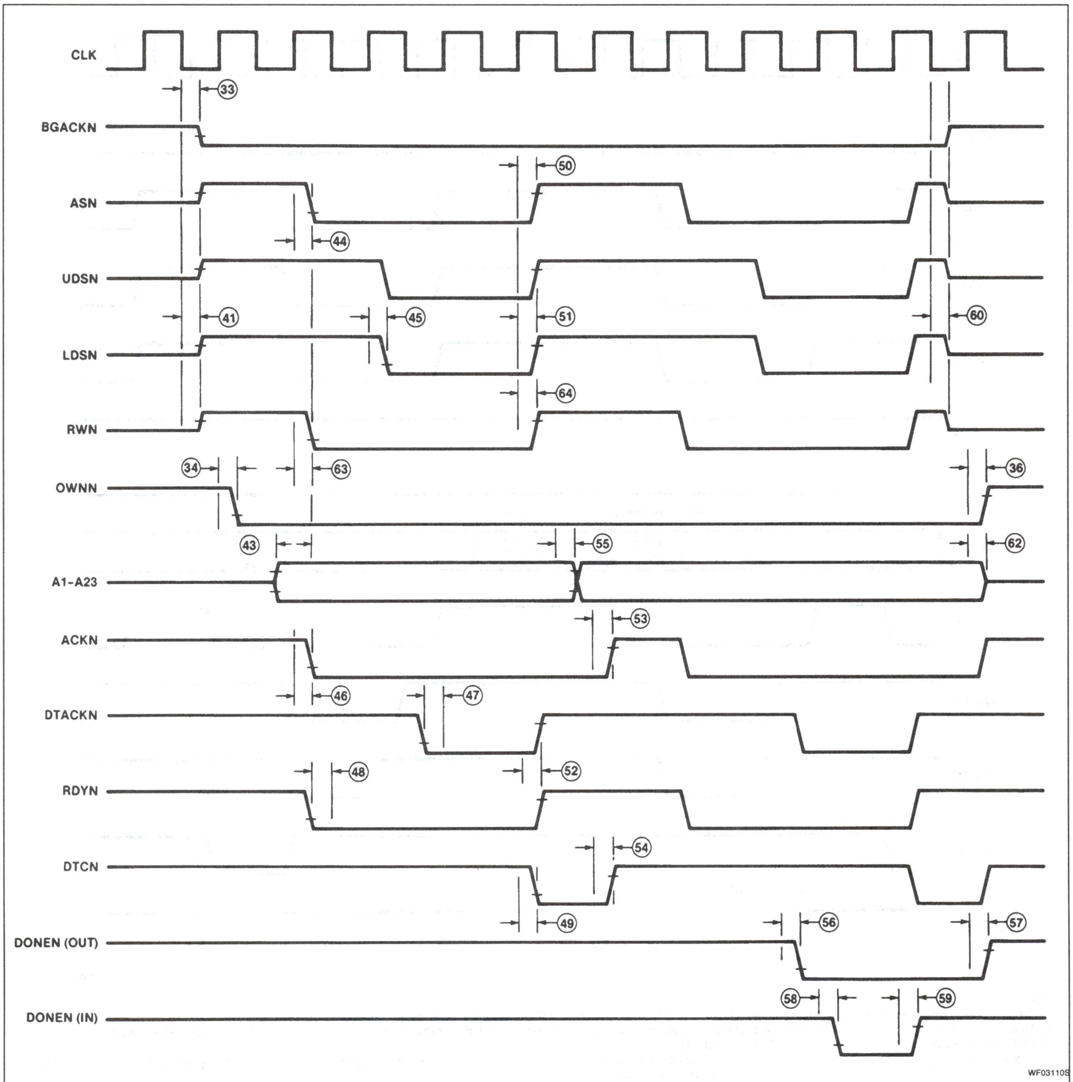
NOTE:
In order to keep the bus, REQN must come no later than the 1/2 clock minus the setup time ㊲ prior to assertion edge of DTCN.

Figure 10. DMAI Cycle Steal Mode Request Timing


CLK
BGACKN
ASN
UDSN
LDSN
RWN
OWNN
A1-A23
ACKN
DTACKN
RDYN
DTCN
DONEN (OUT)
DONEN (IN)
33
44
50
41
45
51
60
34
36
43
55
62
46
53
47
52
48
54
49
56
57
58
59
WF03100S

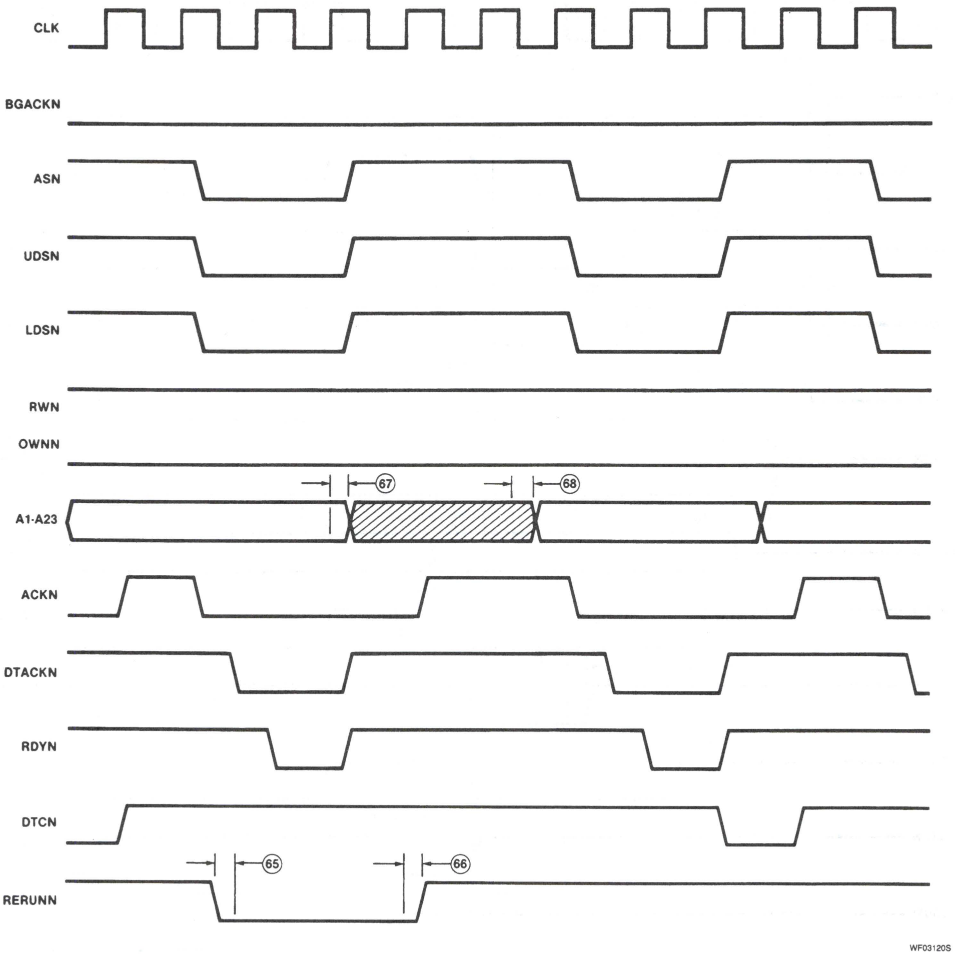
NOTE:
16-bit transfer illustrated. For 8-bit transfer either LDSN or UDSN, but not both, will be asserted each cycle, depending on byte address.

Figure 11. Read From Memory, Write to Device


CLK
BGACKN
ASN
UDSN
LDSN
RWN
OWNN
A1-A23
ACKN
DTACKN
RDYN
DTCN
DONEN (OUT)
DONEN (IN)
33
50
44
41
45
51
60
64
34
63
36
43
55
62
53
46
47
48
52
54
49
56
57
58
59
WF03110S

NOTE:
16-bit transfer illustrated. For 8-bit transfer either LDSN or UDSN, but not both, will be asserted each cycle, depending on byte address.

Figure 12. Write to Memory, Read From Device



NOTES:

1. 16-bit transfer illustrated. For 8-bit transfer either LDSN or UDSN, but not both, will be asserted each cycle, depending on byte address.
2. DMAI will release the bus after a RERUNN if there is no valid request. The next request will then retry the cycle which was terminated by the RERUNN signal.
3. RERUNN must be asserted no later than DTACKN and RDYN.
4. If a cycle is terminated by RERUNN, the transfer count will be one less than the actual data transferred correctly. The double RERUNN signal on the same cycle will terminate the DMAI operation with a status bit set and an interrupt generated (if enabled).

Figure 13. Rerun Asserted During Read From Memory, Write to Device

Signetics

SCN68562
Dual Universal Serial Communications Controller (DUSCC)

Microprocessor Products

Preliminary Specification

DESCRIPTION

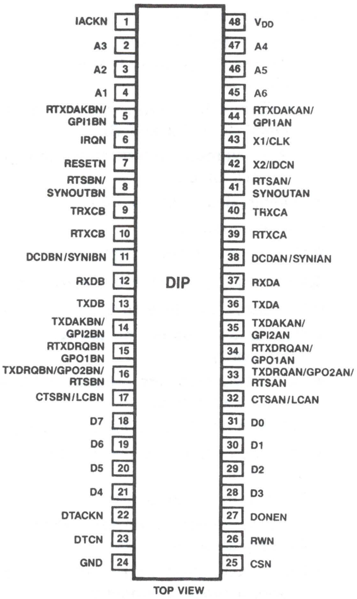
The Signetics SCN68562 Dual Universal Serial Communications Controller (DUSCC) is a single-chip MOS-LSI communications device that provides two independent, multi-protocol, full-duplex receiver/transmitter channels in a single package. It supports bit-oriented and character-oriented (byte count and byte control) synchronous data link controls as well as asynchronous protocols. The SCN68562 interfaces to the 68000 MPU via asynchronous bus control signals and is capable of program-polled, interrupt-driven, block-move or DMA data transfers.

The operating mode and data format of each channel can be programmed independently. Each channel consists of a receiver, a transmitter, a 16-bit multifunction counter/timer, a digital phase-locked loop (DPLL), a parity/CRC generator and checker, and associated control circuits. The two channels share a common bit rate generator (BRG), operating directly from a crystal or an external clock, which provides sixteen common bit rates simultaneously. The operating rate for the receiver and transmitter of each channel can be independently selected from the BRG, the DPLL, the counter/timer, or from an external 1× or 16× clock, making the DUSCC well suited for dual-speed channel applications. Data rates up to 4Mbits are supported.

The transmitter and receiver each contain a four-deep FIFO with appended transmitter command and receiver status bits and a shift register. This permits reading and writing of up to four characters at a time, minimizing the potential of receiver overrun or transmitter underrun, and reducing interrupt or DMA overhead. In addition, a flow control capability is provided to disable a remote transmitter when the FIFO of the local receiving device is full.

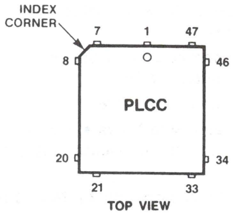
Two modem control inputs (DCD and CTS) and three modem control outputs (RTS and two general purpose) are provided. Because the modem control inputs and outputs are general purpose in nature, they can be optionally programmed for other functions.

PIN CONFIGURATIONS



IACKN 1
A3 2
A2 3
A1 4
RTXDAKBN/GPI1BN 5
IRQN 6
RESETN 7
RTSBN/SYNOUTBN 8
TRXCB 9
RTXCB 10
DCDBN/SYNIBN 11
RXDB 12
TXDB 13
TXDAKBN/GPI2BN 14
RTXDRQBN GPO1BN 15
TXDRQBN/GPO2BN/RTSBN 16
CTSBN/LCBN 17
D7 18
D6 19
D5 20
D4 21
DTACKN 22
DTCN 23
GND 24
48 V_DD
47 A4
46 A5
45 A6
44 RTXDAKAN/GPI1AN
43 X1/CLK
42 X2/IDCN
41 RTSAN/SYNOUTAN
40 TRXCA
39 RTXCA
38 DCDAN/SYNIAN
37 RXDA
36 TXDA
35 TXDAKAN/GPI2AN
34 RTXDRQAN/GPO1AN
33 TXDRQAN/GPO2AN/RTSAN
32 CTSAN/LCAN
31 D0
30 D1
29 D2
28 D3
27 DONEN
26 RWN
25 CSN
DIP
TOP VIEW

CD05090S



INDEX CORNER
7
1
47
8
46
PLCC
20
34
21
33
TOP VIEW

CD0052PS

Pin	Function	Pin	Function
1	IACKN	27	CSN
2	A3	28	R/WN
3	A2	29	DONEN
4	A1	30	D3
5	RTxDAKBN/GPI1BN	31	D2
		32	D1
6	IRQN	33	D0
7	NC	34	NC
8	RESETN	35	CTSAN/LCAN
9	RTSBN/SYNOUTBN	36	TxDRQAN/GPO2AN/RTSAN
10	TRxCB	37	RTxDRQAN/GPO1AN
11	RTxCB		
12	DCDBN/SYNIBN	38	TxDAKAN/GPI2AN
13	NC	39	TxDA
14	RxDB	40	RxDA
15	TxDB	41	NC
16	TxDAKBN/GPI2BN	42	DCDAN/SYNIAN
		43	RTxCA
17	RTxDRQBN/GPO1BN	44	TRxCA
		45	RTSAN/SYNOUTAN
18	TxDRQBN/GPO2BN/RTSBN	46	X2/IDCN
19	CTSBN/LCBN	47	X1/CLK
20	D7	48	RTxDAKAN/GPI1AN
21	D6		
22	D5	49	A6
23	D4	50	A5
24	DTACKN	51	A4
25	DTCN	52	V_{DD}
26	GND		

Dual Universal Serial Communications Controller (DUSCC) SCN68562

FEATURES

General Features

- Dual full-duplex synchronous/asynchronous receiver and transmitter
- Multi-protocol operation
  - BOP: HDLC/ADCCP, SDLC, SDLC loop, X.25 or X.75 link level, e tc.
  - COP: BISYNC, DDCMP, X.21
  - ASYNC: 5 – 8 bits plus optional parity
- Four character receiver and transmitter FIFOs
- 0 to 4MHz data rate
- Programmable bit rate for each receiver and transmitter selectable from:
  - 16 fixed rates: 50 to 38.4k baud
  - One user-defined rate derived from programmable counter/timer
  - External 1× or 16× clock
  - Digital phase-locked loop
- Parity and FCS (frame check sequence LRC or CRC) generation and checking
- Programmable data encoding/decoding: NRZ, NRZI, FM0, FM1, Manchester
- Programmable channel mode: full-half-duplex, auto-echo, or local loopback
- Programmable data transfer mode: polled, interrupt, DMA, wait
- DMA interface
  - Compatible with Signetics' SCB68430 Direct Memory Access Interface (DMAI) and other DMA controllers
  - Half- or full-duplex operation
  - Single or dual address data transfers
  - Automatic frame termination on counter/timer terminal count or DMA DONE
- Interrupt capabilities
  - Daisy chain option
  - Vector output (fixed or modified by status)
  - Programmable internal priorities
  - Maskable interrupt conditions
  - 68000 compatible
- Multi-function programmable 16-bit counter/timer
  - Bit rate generator
  - Event counter
  - Count received or transmitted characters
  - Delay generator
  - Automatic bit length measurement
- Modem controls
  - RTS, CTS, DCD, and up to four general purpose I/O pins per channel
  - CTS and DCD programmable auto-enables for Tx and Rx
  - Programmable interrupt on change of CTS or DCD
- On-chip oscillator for crystal
- TTL compatible
- Single +5V power supply

Asynchronous Mode Features

- Character length: 5 to 8 bits
- Odd or even parity, no parity, or force parity
- Up to two stop bits programmable in 1/16- bit increments
- 1× or 16× Rx and Tx clock factors
- Parity, overrun, and framing error detection
- False start bit detection
- Start bit search ½ bit time after framing error detection
- Break generation with handshake for counting break characters
- Detection of start and end of received break
- Character compare with optional interrupt on match

Character-Oriented Protocol Features

- Character length: 5 to 8 bits
- Odd or even parity, no parity, or force parity
- LRC or CRC generation and checking
- Optional opening PAD transmission
- One or two SYN characters
- External sync capability
- SYN detection and optional stripping
- SYN or MARK linefill on underrun
- Idle in MARK or SYNs
- Parity, FCS, overrun, and underrun error detection
- BISYNC Features
  - EBCDIC or ASCII header, text and control messages
  - SYN, DLE stripping
  - EOM (end of message) detection and transmission
  - Auto transparency mode switching
  - Auto hunt after receipt of EOM sequence (with closing PAD check after EOT or NAK)
  - Control character sequence detection for both transparent and normal text

Bit-Oriented Protocol Features

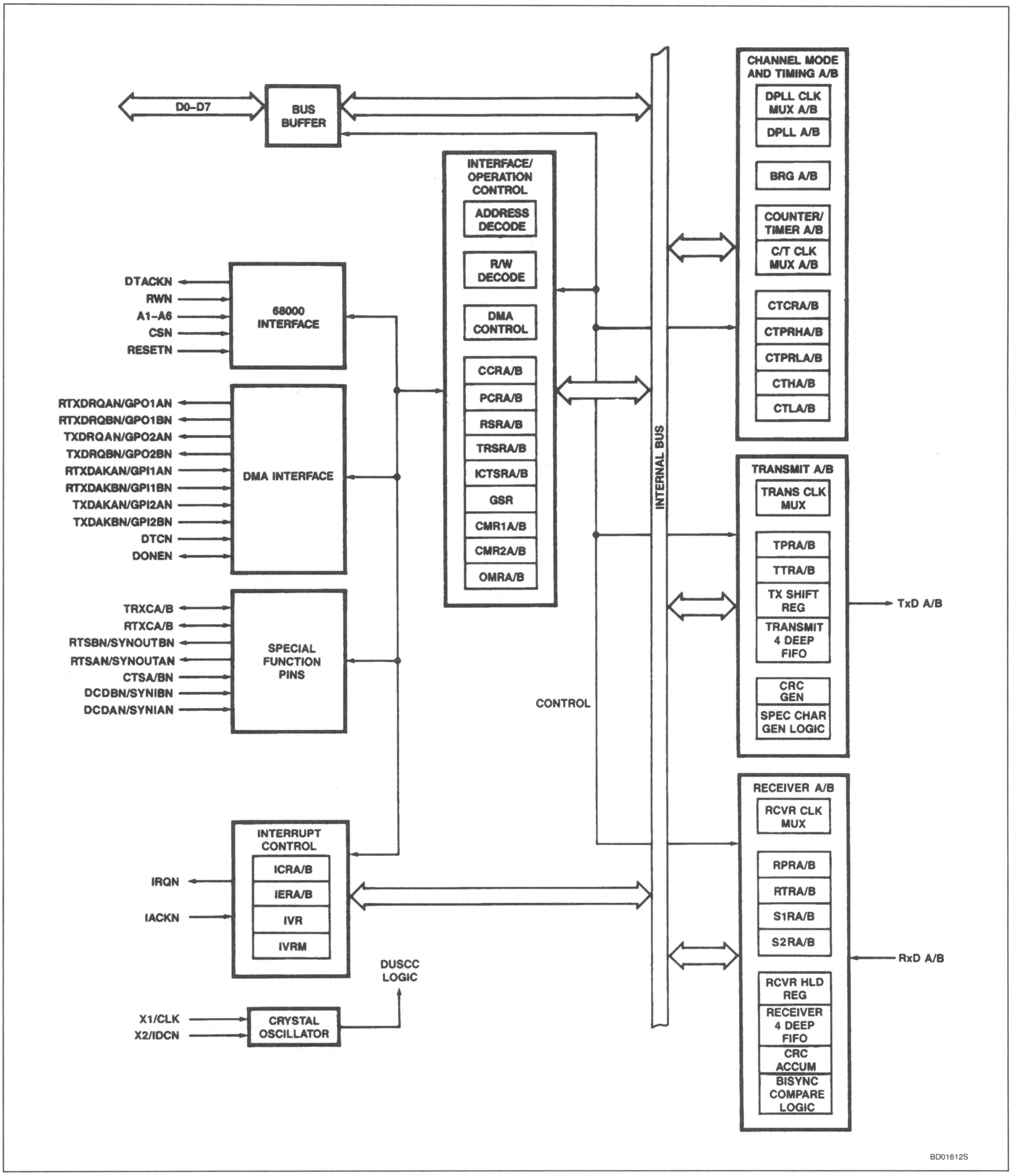
- Character length: 5 to 8 bits
- Detection and transmission of residual character: 0 – 7 bits
- Automatic switch to programmed character length for I field
- Zero insertion and deletion
- Optional opening PAD transmission
- Detection and generation of FLAG, ABORT, and IDLE bit patterns
- Detection and generation of shared (single) FLAG between frames
- Detection of overlapping (shared zero) FLAGs
- ABORT, ABORT-FLAGs, or FCS-FLAGs line fill on underrun
- Idle in MARK or FLAGs
- Secondary address recognition including group and global address
- Single- or dual-octet secondary address
- Extended address and control fields
- Short frame rejection for receiver
- Detection and notification of received end of message
- CRC generation and checking
- SDLC loop mode capability

ORDERING INFORMATION

PACKAGES	V_{CC} = +5V ±5%, T_A = 0 to +70°C
Ceramic DIP	SCN68562C4I48
Plastic DIP	SCN68562C4N48
Plastic LCC	SCN68562C4A52

Dual Universal Serial Communications Controller (DUSCC) SCN68562

BLOCK DIAGRAM


D0–D7
BUS BUFFER
INTERNAL BUS
CHANNEL MODE AND TIMING A/B
DPLL CLK MUX A/B
DPLL A/B
BRG A/B
COUNTER/ TIMER A/B
C/T CLK MUX A/B
CTCRA/B
CTPRHA/B
CTPRLA/B
CTHA/B
CTLA/B
INTERFACE/ OPERATION CONTROL
ADDRESS DECODE
R/W DECODE
DMA CONTROL
CCRA/B
PCRA/B
RSRA/B
TRSRA/B
ICTSRA/B
GSR
CMR1A/B
CMR2A/B
OMRA/B
DTACKN
RWN
A1–A6
CSN
RESETN
68000 INTERFACE
RTXDRQAN/GPO1AN
RTXDRQBN/GPO1BN
TXDRQAN/GPO2AN
TXDRQBN/GPO2BN
RTXDAKAN/GPI1AN
RTXDAKBN/GPI1BN
TXDAKAN/GPI2AN
TXDAKBN/GPI2BN
DTCN
DONEN
DMA INTERFACE
TRXCA/B
RTXCA/B
RTSBN/SYNOUTBN
RTSAN/SYNOUTAN
CTSA/BN
DCDBN/SYNIBN
DCDAN/SYNIAN
SPECIAL FUNCTION PINS
CONTROL
TRANSMIT A/B
TRANS CLK MUX
TPRA/B
TTRA/B
TX SHIFT REG
TRANSMIT 4 DEEP FIFO
CRC GEN
SPEC CHAR GEN LOGIC
TxD A/B
INTERRUPT CONTROL
ICRA/B
IERA/B
IVR
IVRM
IRQN
IACKN
DUSCC LOGIC
X1/CLK
X2/IDCN
CRYSTAL OSCILLATOR
RECEIVER A/B
RCVR CLK MUX
RPRA/B
RTRA/B
S1RA/B
S2RA/B
RxD A/B
RCVR HLD REG
RECEIVER 4 DEEP FIFO
CRC ACCUM
BISYNC COMPARE LOGIC
BD01612S

Dual Universal Serial Communications Controller (DUSCC) SCN68562

PIN DESCRIPTION

In this data sheet, signals are discussed using the terms 'active' and 'inactive' or 'asserted' and 'negated' independent of whether the signal is active in the high (logic 1) or low (logic 0) state. N at the end of a pin name signifies the signal associated with the pin is active low (see individual pin description for the definition of the active level of each signal.) Pins which are provided for both channels are designated by either an underline (_) or by A/B after the name of the pin and the active low state indicator, N, if applicable. A similar method is used for registers provided for both channels; these are designated by either an underline or by A/B after the name.

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
A1 – A6	4 – 2, 45 – 47	5 – 3, 51 – 49	I	**Address Lines:** Active high. Address inputs which specify which of the internal registers is accessed for read/write operations.
D0 – D7	31 – 28, 21 – 18	34 – 31, 23 – 20	I/O	**Bidirectional Data Bus:** Active high, three state. Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the DUSCC take place over this bus. The data bus is enabled when CSN is low, during interrupt acknowledge cycles and single-address DMA acknowledge cycles.
R/WN	26	29	I	**Read/Write:** A high input indicates a read cycle and a low input indicates a write cycle when a cycle is initiated by assertion of the CSN input.
CSN	25	28	I	**Chip Select:** Active low input. When low, data transfers between the CPU and the DUSCC are enabled on D0 – D7 as controlled by the R/WN and A1 – A6 inputs. When CSN is high, the DUSCC is isolated from the data bus (except during interrupt acknowledge cycles and single-address DMA transfers) and D0 – D7 are placed in the tri-state condition.
DTACKN	22	24	O	**Data Transfer Acknowledge:** Active low, 3-State. DTACKN is asserted on a write cycle to indicate that the data on the bus has been latched, and on a read cycle or interrupt acknowledge cycle to indicate valid data is on the bus. The signal is negated when completion of the cycle is indicated by negation of the CSN or IACKN input, and returns to the inactive state (3-State) a short period after it is negated. In single address DMA mode, the operation of this pin is similar to the description above. The exception is that it is negated when completion of the cycle is indicated by the assertion of DTCN or negation of DMA acknowledge inputs (whichever occurs first), and returns to the inactive state (3-State) a short period after it is negated. When negated, DTACKN becomes an open drain output and requires an external pull-up resistor.
IRQN	6	7	O	**Interrupt Request:** Active low, open drain. This output is asserted upon occurrence of any enabled interrupting condition. The CPU can read the general status register to determine the interrupting condition(s), or can respond with an interrupt acknowledge cycle to cause the DUSCC to output an interrupt vector on the data bus.
IACKN	1	2	I	**Interrupt Acknowledge:** Active low. When IACKN is asserted, the DUSCC responds by placing the contents of the interrupt vector register (modified or unmodified by status) on the data bus and asserting DTACKN. If no active interrupt is pending, DTACKN is not asserted.
X1/CLK	43	47	I	**Crystal or External Clock:** When using the crystal oscillator, the crystal is connected between pins X1 and X2. If a crystal is not used, an external clock is supplied at this input. This clock is used to drive the internal bit rate generator, as an optional input to the counter/timer or DPLL, and to provide other required clocking signals.
X2/IDCN	42	46	O	**Crystal or Interrupt Daisy Chain:** Active low. When a crystal is used as the timing source, the crystal is connected between pins X1 and X2. This pin can be programmed to provide an interrupt daisy chain output which propagates the IACKN signal to lower priority devices, if no active interrupt is pending. This pin should be grounded when an external clock is used on X1 and X2, is not used as an interrupt daisy chain output.
RESETN	7	8	I	**Master Reset:** Active low. A low on this pin resets the transmitters and receivers and resets the registers shown in Table 1. Reset is asynchronous, i.e., no clock is required.
RxDA, RxDB	37, 12	41, 13	I	**Channel A (B) Receiver Serial Data Input:** The least significant bit is received first. If external receiver clock is specified for the channel, the input is sampled on the rising edge of the clock.
TxDA, TxDB	36, 13	39, 15	O	**Channel A (B) Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the marking (high) condition when the transmitter is disabled or when the channel is operating in local loopback mode. If external transmitter clock is specified for the channel, the data is shifted on the falling edge of the clock.
RTxCA, RTxCB	39, 10	43, 11	I/O	**Channel A (B) Receiver/Transmitter Clock:** As an input, it can be programmed to supply the receiver, transmitter, counter/timer, or DPLL clock. As an output, can supply the counter/timer output, the transmitter shift clock (1×), or the receiver sampling clock (1×). The maximum external receiver/transmitter clock frequency is 4MHz.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
TRxCA, TRxCB	40, 9	44, 10	I/O	**Channel A (B) Transmitter/Receiver Clock:** As an input, it can supply the receiver, transmitter, counter/timer, or DPLL clock. As an output, it can supply the counter/timer output, the DPLL output, the transmitter shift clock (1×), the receiver sampling clock (1×,), the transmitter BRG clock (16×), the receiver BRG clock (16×), or the internal system clock (X½). The maximum external receiver/transmitter clock frequency is 4MHz.
CTSA/BN, LCA/BN	32, 17	35, 19	I/O	**Channel A (B) Clear-To-Send Input or Loop Control Output:** Active low. The signal can be programmed to act as an enable for the transmitter when not in loop mode. The DUSCC detects logic level transitions on this input and can be programmed to generate an interrupt when a transition occurs. When operating in the BOP loop mode, this pin becomes a loop control output which is asserted and negated by DUSCC commands. This output provides the means of controlling external loop interface hardware to go on-line and off-line without disturbing operation of the loop.
DCDA/BN, SYNIA/BN	38, 11	42, 12	I	**Channel A (B) Data Carrier Detected or External Sync Input:** The function of this pin is programmable. As a DCD active low input, it acts as an enable for the receiver or can be used as a general purpose input. For the DCD function, the DUSCC detects logic level transitions on this input and can be programmed to generate an interrupt when a transition occurs. As an active low external sync input, it is used in COP modes to obtain character synchronization without receipt of a SYN or FLAG character. This mode can be used in disc or tape controller applications or for the optional byte timing lead in X.21.
RTxDRQA/BN, GPO1A/BN	34, 15	37, 17	O	**Channel A (B) Receiver/Transmitter DMA Service Request or General Purpose Output:** Active low. For half-duplex DMA operation, this output indicates to the DMA controller that one or more characters are available in the receiver FIFO (when the receiver is enabled) or that the transmit FIFO is not full (when the transmitter is enabled). For full-duplex DMA operation, this output indicates to the DMA controller that data is available in the receiver FIFO. In non-DMA mode, this pin is a general purpose output that can be asserted and negated under program control.
TxDRQA/BN, GPO2A/BN, RTSA/BN	33, 16	36, 18	O	**Channel A (B) Transmitter DMA Service Request, General Purpose Output, or Request-to-Send:** Active low. For full-duplex DMA operation, this output indicates to the DMA controller that the transmit FIFO is not full and can accept more data. When not in full-duplex DMA mode, this pin can be programmed as a general purpose or a Request-to-Send output, which can be asserted and negated under program control (see Detailed Operation).
RTxDAKA/BN, GPI1A/BN	44, 5	48, 6	I	**Channel A (B) Receiver/Transmitter DMA Acknowledge or General Purpose Input:** Active low. For half-duplex single address DMA operation, this input indicates to the DUSCC that the DMA controller has acquired the bus and that the requested bus cycle (read receiver FIFO or load transmitter FIFO) is beginning. For full-duplex single address DMA operation, this input indicates to the DUSCC that the DMA controller has acquired the bus and that the requested read receiver FIFO bus cycle is beginning. Because the state of this input can be read under program control, it can be used as a general purpose input when not in single address DMA mode.
TxDAKA/BN, GPI2A/BN	35, 14	38, 16	I	**Channel A (B) Transmitter DMA Acknowledge or General Purpose Input:** Active low. When the channel is programmed for full-duplex single address DMA operation, this input is asserted to indicate to the DUSCC that the DMA controller has acquired the bus and that the requested load transmitter FIFO bus cycle is beginning. Because the state of this input can be read under program control, it can be used as a general purpose input when not in full-duplex single address DMA mode.
DTCN	23	25	I	**Device Transfer Complete:** Active low. DTCN is asserted by the DMA controller to indicate that the requested data transfer is complete.
DONEN	27	30	I/O	**Done:** Active low, open drain. See Detailed Operation for a description of the function of this pin.
RTSA/BN, SYNOUTA/BN	41, 8	45, 9	O	**Channel A (B) Sync Detect or Request-to-Send:** Active low. If programmed as a sync output, it is asserted one bit time after the specified sync character (COP or BISYNC modes) or a FLAG (BOP modes) is detected by the receiver. As a Request-to-Send modem control signal, it functions as described previously for the TxDRQ_N/RTS_N pin.
V_{DD}	48	52	I	+5V ±5% power input.
GND	24	26	I	Signal and power ground input.

REGISTERS

The addressable registers of the DUSCC are shown in Table 1. The following rules apply to all registers:

1. A read from a reserved location in the map results in a read from the 'null register'. The null register returns all ones for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle without a write being performed.
2. Unused bits of a defined register are read as zeros, unless ones have been loaded after master reset.
3. Bits that are unused in the chosen mode but are used in others are readable and writable but their contents are ignored in the chosen mode.
4. All registers are addressable as 8-bit quantities. To facilitate operation with the 68000 MOVEP instruction, addresses are ordered such that certain sets of registers may also be accessed as words or long words.

The operation of the DUSCC is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The contents of certain control registers are initialized on RESET (set to zero). Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems, e.g., changing the channel mode at an inappropriate time may cause the reception or transmission of an incorrect character. In general, the contents of registers which control transmitter or receiver operation, or the counter/timer, should be changed only when they are not enabled.

The DUSCC registers can be separated into five groups to facilitate their usage:

1. Channel mode configuration and pin description registers
2. Transmitter and receiver parameter and timing registers
3. Counter/timer control and value registers
4. Interrupt control and status registers
5. Command register

This arrangement is used in the following description of the DUSCC registers.

Channel Mode Configuration and Pin Description Registers

There are five registers in this group for each channel. The bit format for each of these registers is contained in Table 2. The primary function of these registers is to define configuration of the channels and the function of the programmable pins.

Channel Mode Register 1 (CMR1A, CMR1B)

[7:6] Data Encoding — These bits select the data encoding for the received and transmitted data:

00 If the DPLL is set to NRZI mode (see DPLL commands), it selects positive logic (1 = high, 0 = low). If the DPLL is set to FM mode (see DPLL commands), Manchester (bi-phase level) encoding is selected.

01 NRZI. Non-return-to-zero inverted.

10 FM0. Bi-phase space.

11 FM1. Bi-phase mark.

[5] Extended Control (BOP) —

0 No. A one-octet control field follows the address field.

1 Yes. A two-octet control field follows the address field.

[5] Parity (COP/ASYNC), Code Select (BISYNC) —

0 Even parity if with parity is selected by [4:3] or a 0 in the parity bit position if force parity is selected by [4:3]. In BISYNC protocol mode, internal character comparisons are made using EBCDIC coding.

1 Odd parity if with parity is selected by [4:3] or a 1 in the parity bit position if force parity is selected by [4:3]. In BISYNC protocol mode, internal character comparisons are made using 8-bit odd parity ASCII coding. (Note: The receiver should be programmed for 8-bit characters, RPR[1:0] = 11, with no parity, CMR1[4:3] = 00.)

[4:3] Address Mode (BOP) — This field controls whether a single octet or multiple octets follow the opening FLAG(s) for both the receiver and the transmitter. This field is activated by selection of BOP secondary mode through the channel protocol mode bits CMR1_[2:0] (see Detailed Operation).

00 Single-octet address.
01 Extended address.
10 Dual-octet address.
11 Dual-octet address with group.

[4:3] Parity Mode (COP/ASYNC) — This field selects the parity mode for both the receiver and the transmitter. A parity bit is added to the programmed character length if with parity or force parity is selected:

00 No parity. Required when BISYNC protocol mode is programmed.
01 Reserved.
10 With parity. Odd or even parity is selected by [5].
11 Force parity. The parity bit is forced to the state selected by [5].

[2:0] Channel Protocol Mode — This field selects the operational protocol and submode for both the receiver and transmitter:

000 BOP Primary. No address comparison is performed. For receive, all characters received after the opening FLAG(s) are transferred to the FIFO.

001 BOP Secondary. This mode activates the address modes selected by [4:3]. Except in the case of extended address ([4:3]=01), an address comparison is performed to determine if a frame should be received. Refer to Detailed Operation for details of the various addressing modes. If a valid comparison occurs, the receiver is activated and the address octets and all subsequent received characters of the frame are transferred to the receive FIFO.

010 BOP Loop. The DUSCC acts as a secondary station in a loop. The GO-ON-LOOP and GO-OFF-LOOP commands are used to cause the DUSCC to go on and off the loop. Normally, the TxD output echoes the RxD input with a two-bit time delay. If the transmitter is enabled and the 'go active on poll' command has been asserted, the transmitter will begin sending when an EOP sequence consisting of a zero followed by seven ones is detected. The DUSCC changes the last one of the EOP to zero, making it another FLAG, and then operates as described in the detailed operation section. The loop sending status bit (TRSR[6]) is asserted concurrent with the beginning of transmission. The frame should normally be terminated with an EOM followed by an echo of the marking RxD line so that secondary stations further down the loop can append their messages to the messages from up-loop stations by the same process. If the 'go active on poll' command is not asserted, the transmitter remains inactive (other than echoing the received data) even when the EOP sequence is received.

011 BOP Loop without address comparison. Same as normal loop mode except that address field comparisons are disabled. All received frames are transmitted to the CPU.

100 COP Dual SYN. Character sync is achieved upon receipt of a bit sequence matching the contents of the appropriate bits of S1R and S2R (SYN1-SYN2), including parity bits if any.

101 COP Dual SYN (BISYNC). Character sync is achieved upon receipt of a bit

Table 1. DUSCC Register Address Map

ADDRESS BITS[1] 6 5 4 3 2 1	ACRONYM	REGISTER NAME	MODE	AFFECTED BY RESET
c 0 0 0 0 0	CMR1	Channel Mode Register 1	R/W	Yes — 00
c 0 0 0 0 1	CMR2	Channel Mode Register 2	R/W	Yes — 00
c 0 0 0 1 0	S1R	SYN 1/Secondary Address 1 Register	R/W	No
c 0 0 0 1 1	S2R	SYN 2/Secondary Address 2 Register	R/W	No
c 0 0 1 0 0	TPR	Transmitter Parameter Register	R/W	Yes — 00
c 0 0 1 0 1	TTR	Transmitter Timing Register	R/W	No
c 0 0 1 1 0	RPR	Receiver Parameter Register	R/W	Yes — 00
c 0 0 1 1 1	RTR	Receiver Timing Register	R/W	No
c 0 1 0 0 0	CTPRH	Counter/Timer Preset Register High	R/W	No
c 0 1 0 0 1	CTPRL	Counter/Timer Preset Register Low	R/W	No
c 0 1 0 1 0	CTCR	Counter/Timer Control Register	R/W	Yes — 00
c 0 1 0 1 1	OMR	Output and Miscellaneous Register	R/W	Yes — 00
c 0 1 1 0 0	CTH	Counter/Timer High	R	No
c 0 1 1 0 1	CTL	Counter/Timer Low	R	No
c 0 1 1 1 0	PCR	Pin Configuration Register	R/W	Yes — 00
c 0 1 1 1 1	CCR	Channel Command Register	R/W	No
c 1 0 0 X X	TxFIFO	Transmitter FIFO	W	No
c 1 0 1 X X	RxFIFO	Receiver FIFO	R	No
c 1 1 0 0 0	RSR	Receiver Status Register	R/W[2]	Yes — 00
c 1 1 0 0 1	TRSR	Transmitter and Receiver Status Register	R/W[2]	Yes — 00
c 1 1 0 1 0	ICTSR	Input and Counter/Timer Status Register	R/W[2]	Yes
d 1 1 0 1 1	GSR	General Status Register	R/W[2]	Yes — 00
c 1 1 1 0 0	IER	Interrupt Enable Register	R/W	Yes — 00
c 1 1 1 0 1		Not used		
0 1 1 1 1 0	IVR	Interrupt Vector Register — Unmodified	R/W	Yes — 0F
1 1 1 1 1 0	IVRM	Interrupt Vector Register — Modified	R	Yes — 0F
0 1 1 1 1 1	ICR	Interrupt Control Register	R/W	Yes — 00
1 1 1 1 1 1		Not used		

NOTES:

1. c = 0 for channel A, c = 1 for channel B.
   d = don't care — register may be accessed as either channel.
   x = don't care — FIFOs are addressable at any of four adjacent addresses to allow them to be addressed as byte/word/long word with the 68000 MOVEP instruction.
2. A write to this register may perform a status resetting operation.

sequence matching the contents of the appropriate bits of S1R and S2R (SYN1-SYN2). In this mode, special transmitter and receive logic is activated. Transmitter and receiver character length must be programmed to 8 bits and no parity (see Detailed Operation).

110 COP Single SYN. Character sync is achieved upon receipt of a bit sequence matching the contents of the appropriate bits of S1R (SYN1), including parity bit if any. This mode is required when the external sync mode is selected (see description of RPR[4], BOP/COP).

111 Asynchronous. Start/stop format.

Channel Mode Register 2 (CMR2A, CMR2B)

[7:6] Channel Connection — This field selects the mode of operation of the channel. The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character.

00 Normal mode. The transmitter and receiver operate independently in either half- or full-duplex, controlled by the respective enable commands.

01 Automatic echo mode. Automatically retransmits the received data with a half-bit time delay (ASYNC, 16× clock mode) or a two-bit time delay (all other modes). The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxD output.
2. The receiver clock is used for the transmitter for Async 16X clock mode. For other modes the transmitter clock must be supplied.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The TxRDY and underrun status bits are inactive.
5. The received parity and/or FCS are checked if required, but are not regenerated for transmission, i.e., transmitted parity and/or FCS are as received.
6. In ASYNC mode, character framing is checked, but the stop bits are retransmitted as received. A received break is echoed as received.
7. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

10 Local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver if NRZI or NRZ encoding is used. For FM or Manchester encoding because the receiver clock is derived from the DPLL, the DPLL source clock must be maintained.
3. The TxD output is held high.

4. The RxD input is ignored.
5. The receiver and transmitter must be enabled.
6. CPU to transmitter and receiver communications continue normally.

11 Reserved.

[5:3] Data Transfer Interface — This field specifies the type of data transfer between the DUSCC's Rx and TxFIFOs and the CPU. All interrupt and status functions operate normally regardless of the data transfer interface programmed. Refer to Detailed Operation for details of the various DMA transfer interfaces.

000	Half-duplex single address DMA.
001	Half-duplex dual address DMA.
010	Full-duplex single address DMA.
011	Full-duplex dual address DMA.
100	Wait on receive only. In this mode a read of a non-empty receive FIFO results in a normal bus cycle. However, if the receive FIFO of the channel is empty when a read Rx FIFO cycle is initiated, the DTACKN output remains negated until a character is received and loaded into the FIFO. DTACKN is then asserted and the cycle is completed normally.
101	Wait on transmit only. In this mode a write to a non-full transmit FIFO results in a normal bus cycle. However, if the transmit FIFO of the channel is full when a write TxFIFO cycle is initiated, the DTACKN output remains negated until a FIFO position becomes available for the new character. DTACKN is then asserted and the cycle is completed normally.
110	Wait on transmit and receive. As above for both wait on receive and transmit operations.
111	Polled or interrupt. DMA and wait functions of the channel are not activated. Data transfers to the Rx and TxFIFOs are via normal bus read and write cycles in response to polling of the status registers and/or interrupts.

[2:0] Frame Check Sequence Select — This field selects the optional frame check sequence (FCS) to be appended at the end of a transmitted frame. When CRC is selected in COP, then no parity and 8-bit character length must be used. The selected FCS is transmitted as follows:

1. Following the transmission of a FIFOed character tagged with the 'send EOM' command.
2. If underrun control (TPR[7:6]) is programmed for TEOM, upon occurrence of an underrun.
3. If TEOM on zero count or done (TPR[4]) is asserted and the counter/timer is counting transmitted characters, after transmission of the character which causes the counter to reach zero count.
4. In DMA mode with TEOM on zero count or done (TPR[4]) set, after transmission of a character if DONEN is asserted when that character was loaded into the TxFIFO by the DMA controller.

000	No frame check sequence.
001	Reserved
010	LRC8: Divisor = x^8+1, dividend preset to zeros. The Tx sends the calculated LRC non-inverted. The Rx indicates an error if the computed LRC is not equal to 0. Valid for COP modes only.
011	LRC8: Divisor = x^8+1, dividend preset to ones. The Tx sends the calculated LRC non-inverted. The Rx indicates an error if the computed LRC is not equal to 0. Valid for COP modes only.
100	CRC16: Divisor = $x^{16}+x^{15}+x^2+1$, dividend preset to zeros. The Tx sends the calculated CRC non-inverted. The Rx indicates an error if the computed CRC is not equal to 0. Not valid for ASYNC mode.
101	CRC16: Divisor = $x^{16}+x^{15}+x^2+1$, dividend preset to ones. The Tx sends the calculated CRC non-inverted. The Rx indicates an error if the computed CRC is not equal to 0. Not valid for ASYNC mode.
110	CRC-CCITT: Divisor = $x^{16}+x^{12}+x^5+1$, dividend preset to zeros. The Tx sends the calculated CRC non-inverted. The Rx indicates an error if the computed CRC is not equal to 0. Not valid for ASYNC mode.
111	CRC-CCITT: Divisor = $x^{16}+x^{12}+x^5+1$, dividend preset to ones. The Tx sends the calculated CRC inverted. The Rx indicates an error if the computed CRC is not equal to H'F0B8'. Not valid for ASYNC mode.

SYN1/Secondary Address 1 Register (S1RA, S1RB)

[7:0] Character Compare — In ASYNC mode this register holds a 5- to 8-bit long bit pattern which is compared with received characters. If a match occurs, the character compare status bit (RSR[7]) is set. This field is ignored if the receiver is in a break condition.

In COP modes, this register contains the 5- to 8-bit SYN1 bit pattern, right justified. Parity bit need not be included in the value placed in the register even if parity is specified in CMR1[4:3]. However, a character received with parity error, when parity is specified, will not match. In ASYNC or COP modes, if parity is specified, then any unused bits in this register must be programmed to zeros. In BOP secondary mode it contains the address used to compare the first received address octet. The register is not used in BOP primary mode or secondary modes where address comparisons are not made, such as when extended addressing is specified.

SYN2/Secondary Address 2 Register (S2RA, S2RB)

[7:0] — This register is not used in ASYNC, COP single SYN, BOP primary modes, BOP secondary modes with single address field, and BOP secondary modes where address comparisons are not made, such as when extended addressing is specified.

In COP dual SYN modes, it contains the 5- to 8-bit SYN2 bit pattern, right justified. Parity bit need not be included in the value placed in the register even if parity is specified in CMR1[4:3]. However, a character received with parity error, when parity is specified, will not match. If parity is specified, then any unused bits in this register must be programmed to zeros. In BOP secondary mode using two address octets, it contains the partial address used to compare the second received address octet.

Pin Configuration Register (PCRA, PCRB)

This register selects the functions for multipurpose I/O pins.

[7] X2/IDC — This bit is defined only for PCRA. It is not used in PCRB.

0	The X2/IDCN pin is used as a crystal connection.
1	The X2/IDCN pin is the interrupt daisy chain output.

[6] GPO2/RTS — The function of this pin is programmable only when not operating in full-duplex DMA mode.

0	The TxDRQ_N/GPO2_N/RTS_N pin is a general purpose output. It is low when OMR[2] is a 1 and high when OMR[2] is 0.
1	The pin is a request-to-send output (see Detailed Operation).

Table 2. Channel Configuration/Pin Definition Registers Bit Formats

CHANNEL MODE REG 1

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(CMR1A, CMR1B)	**Data Encoding**		**Extended Control**	**Address Mode (BOP)**		**Channel Protocol Mode**		
	00 — NRZ/Manchester 01 — NRZI 10 — FM0 11 — FM1		BOP only 0 — no 1 — yes	00 — 8-bit 01 — extended address 10 — 16-bit 11 — 16-bit w/group		000 — BOP primary 001 — BOP secondary 010 — BOP loop 011 — BOP loop - no adr. comp.		
			#Parity	**Parity Mode (COP/ASYNC)**		100 — COP dual SYN 101 — COP dual SYN (BISYNC) 110 — COP single SYN 111 — asynchronous		
			0 — even 1 — odd	00 — no parity 01 — reserved 10 — with parity 11 — force parity				

NOTE:
#In BISYNC protocol mode, 0 = EBCDIC, 1 = ASCII coding.

CHANNEL MODE REG 2

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(CMR2A, CMR2B)	**Channel Connection**		**Data Transfer Interface**			**Frame Check Sequence Select**		
	00 — normal 01 — auto echo 10 — local loop 11 — reserved		000 — half-duplex single address DMA 001 — half-duplex dual address DMA 010 — full-duplex single address DMA 011 — full-duplex dual address DMA 100 — wait on Rx only 101 — wait on Tx only 110 — wait on Rx or Tx 111 — polled or interrupt			000 — none 001 — reserved 010 — LRC8 preset 0s 011 — LRC8 preset 1s 100 — CRC 16 preset 0s 101 — CRC 16 preset 1s 110 — CRC CCITT preset 0s 111 — CRC CCITT preset 1s		

SYN1/SECONDARY ADDRESS REG 1

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(S1RA, S1RB)	ASYNC — Character compare (5 - 8 bits) COP — SYN1 (5 - 8 bits) BOP — First address octet							

SYN2/SECONDARY ADDRESS REG 2

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(S2RA, S2RB)	ASYNC — not used COP — SYN2 (5 - 8 bits) BOP — Second address octet							

PIN CONFIGURATION REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(PCRA, PCRB)	**X2/IDC**	**GPO2/RTS**	**SYNOUT/RTS**	**RTxC Pin**		**TRxC Pin**		
	* 0 — X2 1 — IDC	0 — GPO2 1 — RTS	0 — SYNOUT 1 — RTS	00 — input 01 — C/T 10 — TxCLK 1× 11 — RxCLK 1×		000 — input 001 — XTAL/2 010 — DPLL 011 — C/T	100 — TxCLK 16× 101 — RxCLK 16× 110 — TxCLK 1× 111 — RxCLK 1×	

NOTE:
*PCRA only. Not used in PCRB.

[5] SYNOUT/RTS —

0 The SYNOUT_N/RTS_N pin is an active low output which is asserted one bit time after a SYN pattern (COP modes) in HSRH/HSRL or FLAG (BOP modes) is detected in CCSR. The output remains asserted for one receiver clock period. See Figure 1 for receiver data path.

1 The pin is a request-to-send output (see Detailed Operation). The logical state of the pin is controlled by OMR[0], when set the output is zero.

[4:3] RTxC —

00 The pin is an input. It must be programmed for input when used as the input for the receiver or transmitter clock, the DPLL, or the C/T.

01 The pin is an output from the counter/timer. Refer to CTCRA/B description.

10 The pin is an output from the transmitter shift register clock.

11 The pin is an output from the receiver shift register clock.

Table 3. Transmitter and Receiver Parameter and Timing Register Bit Format

TRANSMITTER PARAMETER REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 7
(TPRA, TPRB)	**Underrun Control**		**Idle**	**TEOM On Zero Cnt Or Done**	**Tx RTS Control**	**CTS Enable Tx**	**Tx Character Length**	
COP	00 — FCS-idle 01 — reserved 10 — MARKs 11 — SYNs		0 — MARKs 1 — SYNs	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes	00 — 5 bits 01 — 6 bits 10 — 7 bits 11 — 8 bits	
	Underrun Control		**Idle**	**TEOM On Zero Cnt Or Done**				
BOP	00 — FCS-FLAG-idle 01 — reserved 10 — ABORT-MARKs 11 — ABORT-FLAGs		0 — MARKs 1 — FLAGs	0 — no 1 — yes				
	Stop Bits Per Character							
ASYNC	9/16 to 1, 17/16 to 1.5, 25/16 to 2 programmable in 1/16-bit increments							

TRANSMITTER TIMING REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(TTRA, TTRB)	**External Source**	**Transmitter Clock Select**			**Bit Rate Select**			
	0 — RTxC 1 — TRxC	000 — 1× external 001 — 16× external 010 — DPLL 011 — BRG 100 — 2× other channel C/T 101 — 32× other channel C/T 110 — 2× own channel C/T 111 — 32× own channel C/T			one of sixteen rates from BRG			

RECEIVER PARAMETER REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(RPRA, RPRB)	not used	not used	not used	**Rx RTS Control**	**Strip* Parity**	**DCD Enable Rx**	**Rx Character Length**	
ASYNC				0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes	00 — 5 bits 01 — 6 bits 10 — 7 bits 11 — 8 bits	
COP	**SYN Strip**	**FCS to FIFO**	**Auto Hunt & Pad Chk**	**Ext Sync**	**Strip* Parity**			
	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes			
	not used	**FCS to FIFO**	**Overrun Mode**	not used	**All Party Address**			
BOP		0 — no 1 — yes	0 — hunt 1 — cont		0 — no 1 — yes			

NOTE:
*If the receiver character length is 8-bits and parity is programmed, this bit must be set.

Table 3. Transmitter and Receiver Parameter and Timing Register Bit Format (Continued)

RECEIVER TIMING REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(RTRA, RTRB)	**External Source**	**Receiver Clock Select**			**Bit Rate Select**			
	0 — RTxC 1 — TRxC	000 — 1× external 001 — 16× external ⎫ ASYNC 010 — BRG ⎬ protocol 011 - C/T of channel ⎭ mode only 100 — DPLL, source = 64× X1/CLK 101 — DPLL, source = 32× External 110 — DPLL, source = 32× BRG 111 — DPLL, source = 32× C/T			one of sixteen rates from BRG			

OUTPUT AND MISC REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(OMRA, OMRB)	**Tx Residual Character Length**			**TxRDY Activate**	**RxRDY Activate**	**OUT 2**	**OUT 1**	**RTS**
	000 — 1 bit 001 — 2 bits 010 — 3 bits 011 — 4 bits 100 — 5 bits 101 — 6 bits 110 — 7 bits 111 — same as TPR[1:0]			0 — FIFO not full 1 — FIFO empty	0 — FIFO not empty 1 — FIFO full	0 – 0 1 – 1	0 – 0 1 – 1	0 – 0 1 – 1

[2:0] TRxC —

000	The pin is an input. It must be programmed for input when used as the input for the receiver or transmitter clock, the DPLL, or the C/T.
001	The pin is an output from the crystal oscillator. (XTAL/2)
010	The pin is an output from the DPLL output clock.
011	The pin is an output from the counter/timer. Refer to CTCRA/B description.
100	The pin is an output from the transmitter BRG at 16X the rate selected by TTR [3:0].
101	The pin is an output from the receiver BRG at 16X the rate selected by RTR [3:0].
110	The pin is an output from the transmitter shift register clock.
111	The pin is an output from the receiver shift register clock.

Transmitter and Receiver Parameter and Timing Registers

This set of five registers contains the information which controls the operation of the transmitter and receiver for each channel. Table 3 shows the bit map format for each of these registers. The registers of this group are:

1. Transmitter parameter and timing registers (TPRA/B and TTRA/B)
2. Receiver parameter and timing registers (RPRA/B and RTRA/B)
3. Output and miscellaneous register (OMRA/B)

The first and second group of registers define the transmitter and receiver parameters and timing. Included in the receiver timing registers are the programming parameters for the DPLL. The last register of the group, OMR contains additional transmitter and receiver information and controls the logical state of the output pins when they are not used as a part of the channel configuration.

Transmitter Parameter Register (TPRA, TPRB)

[7:6] Underrun Control — In BOP and COP modes, this field selects the transmitter response in the event of an underrun (i.e., the TxFIFO is empty).

00	Normal end of message termination. In BOP, the transmitter sends the FCS (if selected by CMR2[2:0]) followed by a FLAG and then either MARKs or FLAGs, as specified by [5]. In COP, the transmitter sends the FCS (if selected by CMR2[2:0]) and then either MARKs or SYNs, as specified by [5].
01	Reserved.
10	In BOP, the transmitter sends an ABORT (11111111) and then places the TxD output in a marking condition until receipt of further instructions. In COP, the transmitter places the TxD output in a marking condition until receipt of further instructions.
11	In BOP, the transmitter sends an ABORT (11111111) and then sends FLAGs until receipt of further instructions. In COP, the transmitter sends SYNs until receipt of further instructions.

[5] Idle — In BOP and COP modes, this bit selects the transmitter output during idle. Idle is defined as the state following a normal end of message until receipt of the next transmitter command.

0	Idle in marking condition.
1	Idle sending SYNs (COP) or FLAGs (BOP).

[4] Transmit EOM on Zero Count or Done — In BOP and COP modes, the assertion of this bit causes the end of message (FCS in COP, FCS-FLAG in BOP) to be transmitted upon the following events:

1. If the counter/timer is counting transmitted characters, after transmission of the character which causes the counter to reach zero count. (DONEN is also asserted as an output if the channel is in a DMA operation.)
2. If the channel is operating in DMA mode, after transmission of a character if DONEN was asserted when that character was loaded into the TxFIFO by the DMA controller.

[7:4] Stop Bits per Character — In ASYNC mode, this field programs the length of the stop bit appended to the transmitted character as shown in Table 4.

Table 4. Stop Bits — Transmitted Character

[7:4]	5 BITS/ CHAR	6, 7, or 8 BITS/CHAR
0000	1.063	0.563
0001	1.125	0.625
0010	1.188	0.688
0011	1.250	0.750
0100	1.313	0.813
0101	1.375	0.875
0110	1.438	0.938
0111	1.500	1.000
1000	1.563	1.563
1001	1.625	1.625
1010	1.688	1.688
1011	1.750	1.750
1100	1.813	1.813
1101	1.875	1.875
1110	1.938	1.938
1111	2.000	2.000

Stop bit lengths of $\frac{9}{16}$ to 1 and $1\frac{9}{16}$ to 2 bits, in increments of $\frac{1}{16}$- bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, $1\frac{1}{16}$ to 2 stop bits can be programmed in increments of $\frac{1}{16}$- bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1× clock (or a 2× clock for counter/timer) is used for the transmitter, [7] = 0 selects one stop bit and [7] = 1 selects two stop bits to be transmitted. If Manchester, NRZI, or FM data encoding is selected, only integral stop bit lengths should be used.

[3] Transmitter Request-to-Send Control — This bit controls the deactivation of the RTS_N output by the transmitter (see Detailed Operation).

0 RTS_N is not affected by status of transmitter.

1 RTS_N changes state as a function of transmitter status.

[2] Clear-to-Send Enable Transmitter — The state of this bit determines if the CTS_N input controls the operation of the channel's transmitter (see Detailed Operation). The duration of CTS level change is described in the discussion of ICTSR[4].

0 CTS_N has no affect on the transmitter.

1 CTS_N affects the state of the transmitter.

[1:0] Transmitted Bits per Character — This field selects the number of data bits per character to be transmitted. The character length does not include the start, parity, and stop bits in ASYNC or the parity bit in COP. In BOP modes the character length for the address and control fields is always 8 bits, and the value of this field only applies to the information (I) field, except for the last character of the I field, whose length is specified by OMR[7:5].

Transmitter Timing Register (TTRA, TTRB)

[7] External Source — This bit selects the RTxC pin or the TRxC pin of the channel as the transmitter clock input when [6:4] specifies external. When used for input, the selected pin must be programmed as an input in the PCR [4:3] or [2:0].

0 External input from RTxC pin.

1 External input from TRxC pin.

[6:4] Transmitter Clock Select — This field selects the clock for the transmitter.

000 External clock from TRxC or RTxC at 1× the shift (baud) rate.

001 External clock from TRxC or RTxC at 16× the shift rate.

010 Internal clock from the phase locked loop at 1× the bit rate. It should be used only in half-duplex operation since the DPLL will periodically resync itself to the received data if in full-duplex operation.

011 Internal clock from the bit rate generator at 32× the shift rate. The clock signal is divided by two before use in the transmitter which operates at 16× the baud rate. Rate selected by [3:0].

100 Internal clock from counter/timer of other channel. The C/T should be programmed to produce a clock at 2× the shift rate.

101 Internal clock from counter/timer of other channel. The C/T should be programmed to produce a clock at 32× the shift rate.

110 Internal clock from the counter/timer of own channel. The C/T should be programmed to produce a clock at 2× the shift rate.

111 Internal clock from the counter/timer of own channel. The C/T should be programmed to produce a clock at 32× the shift rate.

[3:0] Bit Rate Select — This field selects an output from the bit rate generator to be used by the transmitter circuits. The actual frequency output from the BRG is 32× the bit rate shown in Table 5. With a crystal or external clock of 14.7456MHz the bit rates are as given in Table 5 (this input is divided by two before being applied to the oscillator circuit).

Table 5. Receiver/Transmitter Baud Rates

[3:0]	BIT RATE	[3:0]	BIT RATE
0000	50	1000	1050
0001	75	1001	1200
0010	110	1010	2000
0011	134.5	1011	2400
0100	150	1100	4800
0101	200	1101	9600
0110	300	1110	19.2k
0111	600	1111	38.4k

Receiver Parameter Register (RPRA, RPRB)

[7] SYN Stripping — This bit controls the DUSCC processing in COP modes of SYN 'character patterns' that occur after the initial character synchronization. Refer to Detailed Operation of the receiver for details and definition of SYN 'patterns', and their accumulation of FCS.

0 Strip only leading SYN 'patterns' (i.e. before a message).

1 Strip all SYN 'patterns' (including all odd DLE's in BISYNC transparent mode).

[6] Transfer Received FCS to FIFO — In BISYNC and BOP modes, the assertion of this bit causes the received FCS to be loaded into the RxFIFO. BOP mode operates correctly only if a minimum of two extra FLAGs (without shared zeros) are appended to the frame. If the FCS is specified to be transferred to the FIFO, the EOM status bit will be tagged onto the last byte of the FCS instead of to the last character of the message.

0 Do not transfer FCS to RxFIFO.

1 Transfer FCS to RxFIFO.

[5] Auto-Hunt and Pad Check (BISYNC) — In BISYNC mode, the assertion of this bit causes the receiver to go into hunt for character sync mode after detecting certain end-of-message (EOM) characters. These are defined in the Detailed Operations section for COP receiver operation. After the EOT and NAK sequences, the receiver also does a check for a closing PAD of four 1s.

0 Disable auto-hunt and PAD check.

1 Enable auto-hunt and PAD check.

[5] Overrun Mode (BOP) — The state of this control bit determines the operation of the receiver in the event of a data overrun, i.e., when a character is received while the RxFIFO and the Rx shift register are both full.

0 The receiver terminates receiving the current frame and goes into hunt phase, looking for a FLAG to be received.

1 The receiver continues receiving the current frame. The overrunning character is lost. (The five characters already assembled in the RxFIFO and Rx shift register are protected).

[4] Receiver Request-to-Send Control (ASYNC) — See Detailed Operation.

0 Receiver does not control RTS_N output.

1 Receiver can negate RTS_N output.

[4] External Sync (COP) — In COP single SYN mode, the assertion of this bit enables external character synchronization and receipt of SYN patterns is not required. In order to use this feature, the DUSCC must be programmed to COP single SYN mode, CMR1[2:0] = 110, which is used to set up the internal data paths. In all other respects, however, the external sync mode operation is protocol transparent. A negative signal on the DCD_N/SYNI_N pin will cause the receiver to establish synchronization on the next rising edge of the receiver clock. Character assembly will start at this edge with the RxD input pin considered to have the second bit of data. The sync signal can then be negated. Receipt of the active high external sync input causes the SYN detect status bit (RSR[2]) to be set and the SYNOUT_N pin to be asserted for one bit time. When this mode is enabled, the internal SYN (COP mode) detection and special character recognition (e.g., IDLE, STx, ETx, etc.) circuits are disabled. Character assembly begins as if in the I-field with character length as programmed in RPR[1:0]. Incoming COP frames with parity specified optionally can have it stripped by programming RPR[3]. The user must wait at least eight bit times after Rx is enabled before applying the SYNI_N signal. This time is required to flush the internal data paths. The receiver remains in this mode and further external sync pulses are ignored until the receiver is disabled and then reenabled to resynchronize or to return to normal mode. See Figure 2.

0 External sync not enabled.

1 External sync enabled.

Note that EXT SYNC and DCD ENABLE Rx cannot be asserted simultaneously since they use the same pin.

[3] Strip Parity — In COP and ASYNC modes with parity enabled, this bit controls whether the received parity bit is stripped from the data placed in the receiver FIFO. It is valid only for programmed character lengths of 5, 6, and 7 bits. If the bit is stripped, the corresponding bit in the received data is set to zero. This bit must be set to A '1' if 8-bit character length will parity is programmed.

0 Transfer parity bit as received.

1 Strip parity bit from data.

[3] All Parties Address — In BOP secondary modes, the assertion of this bit causes the receiver to 'wake up' upon receipt of the address H'FF' or H'FF, FF', for single- and dual-octet address modes, respectively, in addition to its normal station address. This feature allows all stations to receive a message.

0 Don't recognize all parties address.

1 Recognize all parties address.

[2] DCD Enable Receiver — If this bit is asserted, the DCD_N/SYNI_N input must be low in order for the receiver to operate. If the input is negated (goes high) while a character is being received, the receiver terminates receipt of the current message (this action in effect disables the receiver). If DCD is subsequently asserted, the receiver will search for the start bit, SYN pattern, or FLAG, depending on the channel protocol. (Note that the change of input can be programmed to generate an interrupt; the duration of the DCD level change is described in the discussion of the input and counter/timer status register ICTSR[5]).

0 DCD not used to enable receiver

1 DCD used to enable receiver

EXT SYNC and DCD ENABLE Rx cannot be asserted simultaneously since they use the same pin.

[1:0] Received Bits per Character — This field selects the number of data bits per character to be assembled by the receiver. The character length does not include the start, parity, and stop bits in ASYNC or the parity bit in COP. In BOP modes, the character length for the address and control fields is always 8 bits, and the value of this field only applies to the information field. If the number of bits assembled for the last character of the I field is less than the value programmed in this field, RCL not zero (RSR[0]) is asserted and the actual number of bits received is given in TRSR[2:0].

Receiver Timing Register (RTRA, RTRB)

[7] External Source — This bit selects the RTxC pin or the TRxC pin of the channel as the receiver or DPLL clock input, when [6:4] specifies external. When used for input, the selected pin must be programmed as an input in the PCR [4:3] or [2:0].

0 External input from RTxC pin.

1 External input from TRxC pin.

[6:4] Receiver Clock Select — This field selects the clock for the receiver.

000 External clock from TRxC or RTxC at 1× the shift (baud) rate.

001 External clock from TRxC or RTxC at 16× the shift rate. Used for ASYNC mode only.

010 Internal clock from the bit rate generator at 32× the shift rate. Clock is divided by two before use by the receiver logic, which operates at 16× the baud rate. Rate selected by [3:0]. Used for ASYNC mode only.

011 Internal clock from counter/timer of own channel. The C/T should be programmed to produce a clock at 32× the shift rate. Clock is divided by two before use in the receiver logic. Used for ASYNC mode only.

100 Internal clock from the digital phase locked loop. The clock for the DPLL is a 64× clock from the crystal oscillator or system clock input. (The input to the oscillator is divided by two).

101 Internal clock from the digital phase locked loop. The clock for the DPLL is an external 32× clock from the RTxC or TRxC pin, as selected by [7].

110 Internal clock from the digital phase locked loop. The clock for the DPLL is a 32× clock from the BRG. The frequency is programmed by [3:0].

111 Internal clock from the digital phase locked loop. The clock for the DPLL is a 32× clock from the counter/timer of the channel.

[3:0] Bit Rate Select — This field selects an output from the bit rate generator to be used by the receiver circuits. The actual frequency output from the BRG is 32× the bit rate shown in Table 5.

Output and Miscellaneous Register (OMRA, OMRB)

[7:5] Transmitted Residual Character Length — In BOP modes, this field determines the number of bits transmitted for the last character in the information field. This length applies to:

- the character in the transmit FIFO accompanied by the FIFOed TEOM command.
- the character loaded into the FIFO by the DMA controller if DONEN is simultaneously asserted and TPR[4] is asserted.
- the character loaded into the FIFO which causes the counter to reach zero count when TPR[4] is asserted.

The length of all other characters in the frame's information field is selected by TPR[1:0]. If this field is 111, the number of bits in the last character is the same as programmed in TPR[1:0].

[4] TxRDY Activate Mode —

0 FIFO not full. The channel's TxRDY status bit is asserted each time a character is transferred from the transmit FIFO to the transmit shift register. If not reset by the CPU, TxRDY remains asserted until the FIFO is full, at which time it is automatically negated.

1 FIFO empty. The channel's TxRDY status bit is asserted when a character transfer from the transmit FIFO to the transmit shift register causes the FIFO to become empty. If not reset by the CPU, TxRDY remains asserted until the FIFO is full, at which time it is negated.

If the TxRDY status bit is reset by the CPU, it will remain negated regardless of the current state of the transmit FIFO, until it is asserted again due to the occurrence of one of the above conditions.

[3] RxRDY Activate Mode —

0 FIFO not empty. The channel's RxRDY status bit is asserted each time a character is transferred from the receive shift register to the receive FIFO. If not reset by the CPU, RxRDY remains asserted until the receive FIFO is empty, at which time it is automatically negated.

1 FIFO full. The channel's RxRDY status bit is asserted when a character transfer from the receive shift register to the receive FIFO causes the FIFO to become full. If not reset by the CPU, RxRDY remains asserted until the FIFO is empty, at which time it is negated.

The RxRDY status bit will also be asserted, regardless of the receiver FIFO full condition, when an end-of-message character is loaded in the RxFIFO (BOP/BISYNC), when a BREAK condition (ASYNC mode) is detected in RSR[2], or when the counter/timer is programmed to count received characters and the character which causes it to reach zero is loaded in the FIFO (all modes). (Refer to the detailed operation of the receiver.)

If reset by the CPU, the RxRDY status bit will remain negated, regardless of the current state of the receiver FIFO, until it is asserted again due to one of the above conditions.

[2] General Purpose Output 2 — This general purpose bit is used to control the TxDRQ_N/GP02_/RTS_N pin, when it is used as an output. The output is high when the bit is a 0 and is low when the bit is a 1.

[1] General Purpose Output 1 — This bit is used to control the RTxDRQ_N/GPO1_N output, which is a general purpose output when the channel is not in DMA mode. The output is high when the bit is a 0 and is low when the bit is a 1.

[0] Request-to-Send Output — This bit controls the TxDRQ_N/GPO2_N/RTS_N and SYNOUT_N/RTS_N pin, when either is used as a RTS output. The output is high when the bit is a 0 and is low when the bit is a 1.

Counter/Timer Control and Value Registers

There are five registers in this set consisting of the following:

1. Counter/timer control register (CTCRA/B)
2. Counter/timer preset high and low registers (CTPRHA/B, CTPRLA/B)
3. Counter/timer (current value) high and low registers (CTHA/B, CTLA/B)

The format of each of the registers of this set is contained in Table 6. The control register contains the operational information for the counter/timer. The preset registers contain the count which is loaded into the counter/timer circuits. The third group contains the current value of the counter/timer as it operates.

Counter/Timer Control Register (CTCRA, CTCRB)

[7] Zero Detect Interrupt — This bit determines whether the assertion of the C/T ZERO COUNT status bit (ICTSR[6]) causes an interrupt to be generated.

0 Interrupt disabled.

1 Interrupt enabled if master interrupt enable (ICR[1] or ICR[0]) is asserted.

[6] Zero Detect Control — This bit determines the action of the counter upon reaching zero count.

0 The counter/timer is preset to the value contained in the counter/timer preset registers (CTPRL, CTPRH) at the next clock edge.

1 The counter/timer continues counting without preset. The value at the next clock edge will be H'FFFF'.

[5] Counter/Timer Output Control — This bit selects the output waveform when the counter/timer is selected to be output on TRxC or RTxC.

1 The output is a single clock positive width pulse each time the C/T reaches zero count. (The duration of this pulse is one clock period.)

0 The output toggles each time the C/T reaches zero count. The output is cleared to low by either of the preset counter/timer commands.

[4:3] Clock Select — This field selects whether the clock selected by [2:0] is prescaled prior to being applied to the input of the C/T.

00	No prescaling.
01	Divide clock by 16.
10	Divide clock by 32.
11	Divide clock by 64.

[2:0] Clock Source — This field selects the clock source for the counter timer.

000	RTxC pin. Pin must be programmed as input.
001	TRxC pin. Pin must be programmed as input.
010	Source is the crystal oscillator or system clock input divided by four.
011	This selects a special mode of operation. In this mode the counter, after receiving the 'start C/T' command, delays the start of counting until the RxD input goes low. It continues counting until the RxD input goes high, then stops and sets the C/T zero count status bit. The CPU can use the value in the C/T to determine the bit rate of the incoming data. The clock is the crystal oscillator or system clock input divided by four.
100	Source is the 32× BRG output selected by RTR[3:0] of own channel.
101	Source is the 32× BRG output selected by TTR[3:0] of own channel.
110	Source is the internal signal which loads received characters from the receive shift register into the receiver FIFO. When operating in this mode, the FIFOed EOM status bit (RSR[7]) shall be set when the character which causes the count to go to zero is loaded into the receive FIFO.
111	Source is the internal signal which transfers characters from the data bus into the transmit FIFO. When operating in this mode, and if the TEOM on zero count or done control bit (TPR[4]) is asserted, the FIFOed Send EOM command will be automatically asserted when the character which causes the count to go to zero is loaded into the transmit FIFO.

Counter/Timer Preset High Register (CTPRHA, CTPRHB)

[7:0] MSB — This register contains the eight most significant bits of the value loaded into

Table 6. Counter/Timer Control and Value Register Bit Formats

COUNTER/TIMER CONTROL REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(CTCRA, CTCRB)	**Zero Detect Interrupt**	**Zero Detect Control**	**Output Control**	**Prescaler**		**Clock Source**		
	0 — disable 1 — enabled	0 — preset 1 — continue	0 — square 1 — pulse	00 — 1 01 — 16 10 — 32 11 — 64		000 — RTxC pin 001 — TRxC pin 010 — X1/CLK divided by 4 011 — X1/CLK divided by 4 gated by RxD 100 — Rx BRG 101 — Tx BRG 110 — Rx characters 111 — Tx characters		

COUNTER/TIMER PRESET HIGH REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(CTPRHA, CTPRHB)	Most significant bits of counter/timer preset value							

COUNTER/TIMER PRESET REGISTER LOW

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(CTPRLA, CTPRLB)	Least significant bits of counter/timer preset value							

COUNTER/TIMER HIGH

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(CTHA, CTHB)	Most significant bits of counter/timer							

COUNTER/TIMER LOW

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(CTLA, CTLB)	Least significant bits of counter/timer							

the counter/timer upon receipt of the load C/T from preset register command or when the counter/timer reaches zero count and the zero detect control bit (CTCR[6]) is negated. The minimum 16-bit counter/timer preset value is H'0002'.

Counter/Timer Preset Low Register (CTPRLA, CTPRLB)

[7:0] LSB — This register contains the eight least significant bits of the value loaded into the counter/timer upon receipt of the load C/T from preset register command or when the counter/timer reaches zero count and the zero detect control bit (CTCR[6]) is negated. The minimum 16-bit counter/timer preset value is H'0002'.

Counter/Timer High Register (CTHA, CTHB)

[7:0] MSB — A read of this 'register' provides the eight most significant bits of the current value of the counter/timer. It is recommended that the C/T be stopped via a stop counter command before it is read in order to prevent errors which may occur due to the read being performed while the C/T is changing. This count is continued after the register is read.

Counter/Timer Low Register (CTLA, CTLB)

[7:0] LSB — A read of this 'register' provides the eight least significant bits of the current value of the counter/timer. It is recommended that the C/T be stopped via a stop counter command before it is read, in order to prevent errors which may occur due to the read being performed while the C/T is changing. This count is continued after the register is read.

Interrupt Control and Status Registers

This group of registers define mechanisms for communications between the DUSCC and the processor and contain the device status information. Four registers, available for each channel, and four common device registers comprise this group which consists of the following:

1. Interrupt enable register (IERA/B)
2. Receiver status register (RSRA/B)
3. Transmitter and receiver status register (TRSRA/B)

4. Input and counter/timer status register (ICTSRA/B)
5. Interrupt vector register (IVR) and modified interrupt vector register (IVRM)
6. Interrupt control register (ICR)
7. General status register (GSR)

See Table 7 for bit formats and Figure 3 for table relationships.

Interrupt Enable Register (IERA, IERB)
This register controls whether the assertion of bits in the channel's status registers causes an interrupt to be generated. An additional condition for an interrupt to be generated is that the channel's master interrupt enable bit, ICR[0] or ICR[1], be asserted.

[7] DCD/CTS —
0 Interrupt not enabled.
1 Interrupt generated if ICTSR[4] or ICTSR[5] are asserted.

[6] TxRDY —
0 Interrupt not enabled.
1 Interrupt generated if TxRDY (GSR[1] or GSR[5] for channels A and B respectively) is asserted.

[5] TRSR 73 —
0 Interrupt not enabled.
1 Interrupt generated if bits 7, 6, 5, 4 or 3 of the TRSR are asserted.

[4] RxRDY —
0 Interrupt not enabled.
1 Interrupt generated if RxRDY (GSR[0] or GSR[4] for channels A and B respectively) is asserted.

[3] RSR 76 —
0 Interrupt not enabled.
1 Interrupt generated if bits 7 or 6 of the RSR are asserted.

[2] RSR 54 —
0 Interrupt not enabled.
1 Interrupt generated if bits 5 or 4 of the RSR are asserted.

[1] RSR 32 —
0 Interrupt not enabled.
1 Interrupt generated if bits 3 or 2 of the RSR are asserted.

Table 7. Interrupt Control and Status Register Bit Format

RECEIVER STATUS REG

	*BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(RSRA, RSRB) ASYNC	# Char compare	RTS negated	Overrun error	not used	BRK end detect	BRK start detect	# Framing error	# Parity error
COP	# EOM detect +	PAD error +	Overrun error	not used	not used	Syn detect	# CRC error	# Parity error
BOP	# EOM detect	Abort detect	Overrun error	Short frame detect	Idle detect	Flag detect	# CRC error	# RCL not zero
LOOP	# EOM detect	Abort/EOP detect	Overrun error	Short frame detect	Turn-around detect	Flag detect	# CRC error	# RCL not zero

NOTES:
Status bit is FIFOed.
+ COP BISYNC mode only.
* All modes indicate character count complete.

TRANSMITTER AND RECEIVER STATUS REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(TRSRA, TRSRB) ASYNC	Transmitter empty	CTS underrun	not used	Send break ack	DPLL error	not used	not used	not used
COP	Transmitter empty	CTS underrun	Frame complete	Send SOM ack	DPLL error	not used	Rx hunt mode	Rx xpnt mode
BOP	Transmitter empty	CTS underrun Loop sending*	Frame complete	Send SOM/ abort ack	DPLL error	**Rx Residual Character Length** 000 — 0 bit 001 — 1 bits 010 — 2 bits 011 — 3 bits 100 — 4 bits 101 — 5 bits 110 — 6 bits 111 — 7 bits		

NOTE:
*Loop mode only

INPUT AND COUNTER/TIMER STATUS REGISTER

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(ICTSRA, ICTSRB)	C/T running	C/T zero count	Delta DCD	Delta CTS/LC	DCD	CTS/LC	GPI2	GPI1

Table 7. Interrupt Control and Status Register Bit Format (Continued)

INTERRUPT ENABLE REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(IERA, IERB)	**DCD/CTS**	**TxRDY**	**TRSR [7:3]**	**RxRDY**	**RSR [7:6]**	**RSR [5:4]**	**RSR [3:2]**	**RSR [1:0]**
	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes

INTERRUPT VECTOR REG AND INTERRUPT VECTOR MODIFIED REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(IVR, IVRM)	8-bit interrupt vector							

GENERAL STATUS REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(GSR)	**Channel B**				**Channel A**			
	External or C/T status	Rx/Tx status	TxRDY	RxRDY	External or C/T status	Rx/Tx status	TxRDY	RxRDY

INTERRUPT CONTROL REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(ICR)	**Channel A/B Interrupt Priority**		**Vector Mode**		**Bits to Modify**	**Vector Includes Status**	**Channel A Master Int Enable**	**Channel B Master Int Enable**
	00 — channel A 01 — channel B 10 — interleaved A 11 — interleaved B		00 — vectored 01 — vectored 10 — vectored 11 — non vectored		0 — 2:0 1 — 4:2	0 — no 1 — yes	0 — no 1 — yes	0 — no 1 — yes

[0] RSR 10 —

0 Interrupt not enabled.

1 Interrupt generated if bits 1 or 0 of the RSR are asserted.

Receiver Status Register (RSRA, RSRB)
This register informs the CPU of receiver status. Bits indicated as 'not used' in a particular mode will read as zero. The logical OR of these bits is presented in GSR[2] or GSR[6] (ORed with the bits of TRSR) for channels A and B, respectively. Unless otherwise indicated, asserted status bits are reset only by performing a write operation to the status register with the bits to be reset being ones in the accompanying data word, or when the RESETN input is asserted, or when a 'reset receiver' command is issued.

Certain status bits are specified as being FIFOed. This means that they occupy positions in a status FIFO that correspond to the data FIFO. As the data is brought to the top of the FIFO (the position read when the RxFIFO is read), the FIFOed status bits are logically ORed with the previous contents of the corresponding bits in the status register. This permits the user to obtain status either character by character or on a block basis. For character by character status, the SR bits should be read and then cleared before reading the character data from RxFIFO. For block status, the status register is initially cleared and then read after the message is received. Asserted status bits can be programmed to generate an interrupt (see Interrupt Enable Register).

[7] Character Count Complete (All Modes), Character Compare (ASYNC), EOM (BISYNC/BOP/LOOP) — If the counter/timer is programmed to count received characters, this bit is asserted when the character which causes the count to go to zero is loaded into the receive FIFO. It is also asserted to indicate the following conditions:

ASYNC The character currently at the top of RxFIFO matched the contents of S1R. A character will not compare if it is received with parity error even if the data portion matches.

BISYNC The character currently at the top of the FIFO was either a text message terminator or a control sequence received outside of a text or header field. See Detailed Operation of COP Receiver. If transfer FCS to FIFO (RPR[6]) is set, the EOM will instead be tagged onto the last byte of the FCS. Note that if an overrun occurs during receipt of a message, the EOM character may be lost, but this status bit will still be asserted to indicate that an EOM was received. For two-byte EOM comparisons, only the second byte is tagged (assuming the CRC is not transferred to the FIFO).

BOP, LOOP The character currently at the top of the FIFO was the last character of the frame. If transfer FCS to FIFO (RPR[6]) is asserted, the EOM will be tagged instead onto the last byte of the FCS. Note that if an overrun occurs, the EOM character may be lost, but this status bit will still be asserted to indicate that an EOM was received. This bit will not be set when an abort is received.

[6] RTS Negated (ASYNC), PAD Error (BISYNC), ABORT (BOP) —

ASYNC	The RTSN output was negated due to receiving the start bit of a new character while the RxFIFO was full (see RPR[4]).
BISYNC	PAD error detected (see RPR[5]).
BOP	An ABORT sequence consisting of a zero followed by seven ones was received after receipt of the first address octet but before receipt of the closing FLAG. The user should read RxFIFO until it is empty and determine if any valid characters from a previous frame are in the FIFO. If no character with a tagged EOM detect ([7]) is found, all characters are from the current frame and should be discarded along with any previously read by the CPU. An ABORT detect causes the receiver to automatically go into search for FLAG state. An abort during a valid frame does not cause the CRC to reset; this will occur when the next frame begins.
LOOP	Performs the ABORT detect function as described for BOP without the restriction that the pattern be detected during an active frame. A zero followed by seven ones is the end-of-poll sequence which allows the transmitter to go active if the 'go active on poll' command has been invoked.

[5] Overrun Error (All Modes) — A new character was received while the receive FIFO was full and a character was already waiting in the receive shift register to be transferred to the FIFO. The DUSCC protects the five characters previously assembled (four in RxFIFO, one in the Rx shift register) and discards the overrunning character(s). After the CPU reads the FIFO, the character waiting in the RxSR will be loaded into the available FIFO position. This releases the RxSR and a new character assembly will start at the next character boundary. In this way, only valid characters will be assembled, i.e. no partial character assembly will occur regardless of when the RxSR became available during the incoming data stream.

[4] Short Frame (BOP/LOOP) —

ASYNC	Not used
COP	Not used
BOP, LOOP	A closing flag was received with missing fields in the frame. See detailed operation for BOP receiver.

[3] BREAK End Detect (ASYNC), IDLE (BOP), Turnaround (LOOP) —

ASYNC	1× clock mode: The RxD input has returned to the marking state for at least one period of the 1× receiver clock after detecting a BREAK. 16× clock mode: The RxD input has returned to the marking (high) state for at least one-half bit time after detecting a BREAK. A half-bit time is defined as eight clock cycles of the 16× receiver clock.
COP	Not used
BOP	An IDLE sequence consisting of a zero followed by fifteen ones was received. During a valid frame, an abort must precede an idle. However, outside of a valid frame, an idle is recognized and abort is not.
LOOP	A turnaround sequence consisting of eight contiguous zeros was detected outside of an active frame. This should normally be used to terminate transmitter operation and return the system to the 'echoing RxD' mode.

[2] BREAK Start Detect (ASYNC), SYN Detect (COP), FLAG Detect (BOP/LOOP)

ASYNC	An all zero character, including parity (if specified) and first stop bit, was received. The receiver shall be capable of detecting breaks which begin in the middle of a previous character. Only a single all-zero character shall be put into the FIFO when a break is detected. Additional entries to the FIFO are inhibited until the end of break has been detected (see above) and a new character is received.
COP	A SYN pattern was received. Refer to Detailed Operation for definition of SYN patterns. Set one bit time after detection of SYN pattern in HSRH, HSRL. See Figure 1 for receiver data path.
BOP, LOOP	A FLAG sequence (01111110) was received. Set one bit time after FLAG is detected in CCSR. See Figure 1 for receiver data path.

[1] Framing Error (ASYNC), CRC Error (COP/BOP/LOOP) —

ASYNC	At the first stop bit position the RxD input was in the low (space) state. The receiver only checks for framing error at the nominal center of the first stop bit regardless of the number of stop bits programmed in TPR[7:4]. This bit is not set for BREAKS.
COP	In BISYNC COP mode, this bit is set upon receipt of the BCC byte(s), if any, to indicate that the received BCC was in error. The bit is normally FIFOed with the last byte of the frame (the character preceding the first BCC byte). However, if transfer FCS to FIFO (RPR[6]) is asserted, this bit is FIFOed with the last BCC byte. The value of this bit should be ignored for non-text messages or if the received frame was aborted via an ENQ. In non-BISYNC COP modes, the bit is set with each received character if the current value of the CRC checker is not equal to the non-error value (see CMR2[2:0]).
BOP, LOOP	This bit is set upon receipt of the FCS byte(s), if any, to indicate that the received FCS was in error. The bit is normally FIFOed with the last byte of the I field (the character preceding the first FCS byte). However, if transfer FCS to FIFO (RPR[6]) is asserted, this bit is FIFOed with the last FCS byte.

[0] Parity Error (ASYNC/COP), RCL Not Zero (BOP/LOOP) —

ASYNC	The parity bit of the received character was not as expected. A parity error does not affect the parity bit put into the FIFO as part of the character when strip parity (RPR [3]) is negated.
COP	The parity bit of the received character was not as expected. A parity error does not affect the parity bit put into the FIFO as part of the character when strip parity (RPR[3]) is negated. A SYN or other character received with parity error is treated as a data character. Thus, a SYN with parity error received while in SYN search state will not establish character sync. Characters received with parity error while in the SYN search state will not set the error bit.
BOP, LOOP	The last character of the I field did not have the character length specified in RPR[1:0]. The actual received character length of this byte can be read in TRSR[2:0]. This bit is FIFOed with the EOM character but TRSR[2:0] is not. An exception occurs if the command to transfer the FCS to the FIFO is active. In this case, the bit will be FIFOed with the last byte of the FCS, i.e., with REOM. In the event

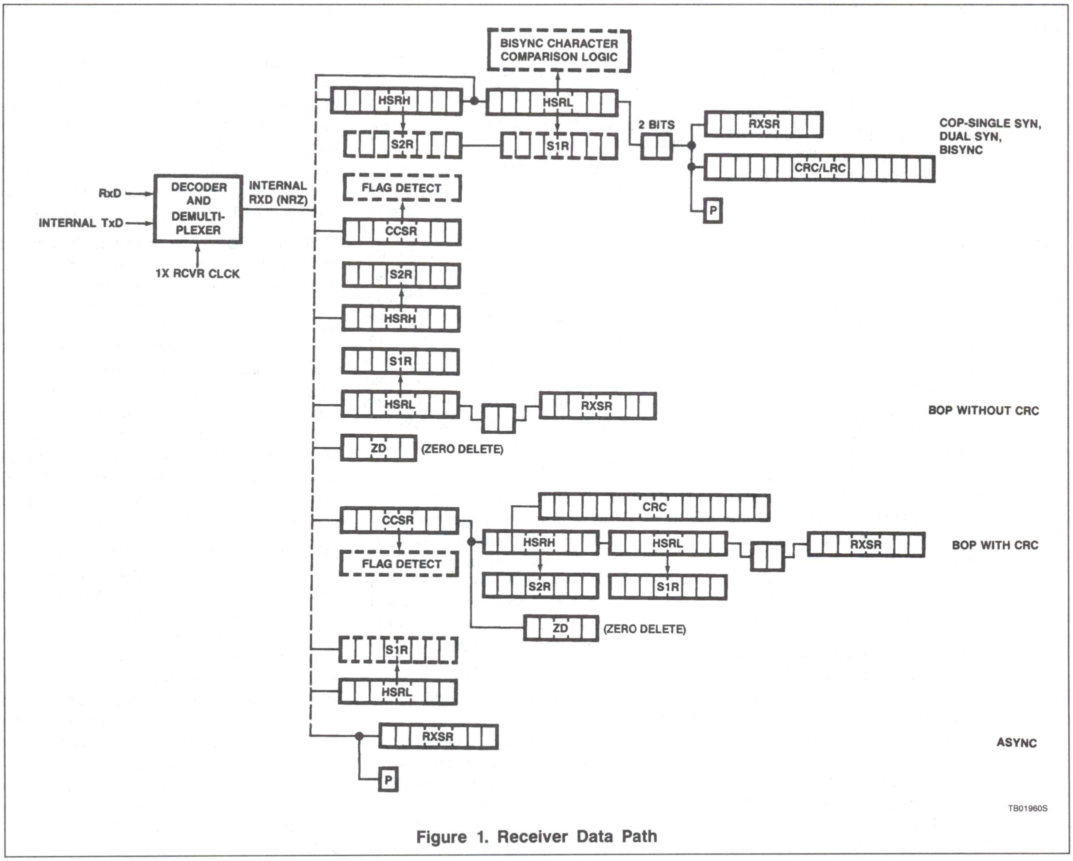

BISYNC CHARACTER COMPARISON LOGIC
HSRH
HSRL
S2R
S1R
2 BITS
RXSR
CRC/LRC
P
COP-SINGLE SYN, DUAL SYN, BISYNC
RxD
INTERNAL TxD
DECODER AND DEMULTI-PLEXER
INTERNAL RXD (NRZ)
1X RCVR CLCK
FLAG DETECT
CCSR
S2R
HSRH
S1R
HSRL
RXSR
BOP WITHOUT CRC
ZD
(ZERO DELETE)
CCSR
CRC
HSRH
HSRL
RXSR
BOP WITH CRC
FLAG DETECT
S2R
S1R
ZD
(ZERO DELETE)
S1R
HSRL
RXSR
P
ASYNC
TB01960S

Figure 1. Receiver Data Path

that residual characters from two consective frames are received and are both in the FIFO, the length in TRSR[2:0] applies to the last received residual character.

Transmitter/Receiver Status Register (TRSRA, TRSRB)

This register informs the CPU of transmitter and receiver status. Bits indicated as not used in a particular mode will read as zero, except for bits [2:0], which may not be zero. The logical-OR of bits [7:3] is presented in GSR[2] or GSR[6] (ORed with the bits of RSR) for channels A and B, respectively. Unless otherwise indicated, asserted status bits are reset only:

1. By performing a write operation to the status register with the bits to be reset being ones in the accompanying data word [7:3].
2. When the RESETN input is asserted.
3. For [7:4], when a 'reset transmitter' command is issued.
4. For [3:0], when a 'reset receiver' command is issued.
5. For [2:0], see description in BOP mode.

Asserted status bits in [7:3] can be programmed to generate an interrupt. See IER.

[7] Transmitter Empty — Indicates that the transmit shift register has completed serializing a character and found no other character to serialize in the TxFIFO. The bit is not set until at least one character from the transmit FIFO (not including PAD characters in synchronous modes) has been serialized. The transmitter action after transmitter empty depends on operating mode:

ASYNC	The TxD output is held in the MARK state until another character is loaded into the TxFIFO. Normal operation then continues.
COP	Action is specified by TPR[7:6].
BOP, LOOP	Action is specified by TPR[7:6].

[6] CTS Underrun (ASYNC/COP/BOP), Loop sending (LOOP) —

ASYNC, COP, BOP	This bit is set only if CTS enable Tx (TPR [2]) is asserted. It indicates that the transmit shift register was ready to begin serializing a character and found the CTS_N input negated. In ASYNC mode, this bit will be reasserted if cleared by the CPU while the CTS_N input is negated.
LOOP	Asserted when the go active on poll command has been invoked and an EOP sequence has been detected, causing the transmitter to go active by changing the EOP to a FLAG (see detailed operation of transmitter).

[5] Frame Complete (COP/BOP) —

ASYNC	Not used.
COP	Asserted at the beginning of transmission of the end of message sequence invoked by which is either a TEOM command, or when TPR[4], or TPR[7:6] = 00. The CPU can invoke the TSOM command after this bit is set to control the number of SYNs between transmitted frames.
BOP	Asserted at the beginning of transmission of the end of message sequence which is invoked by either a TEOM command, or when TPR[4] = 1, or TPR[7:6] = 00. The CPU can invoke the TSOM command after this bit is set to control the number of FLAGs between transmitted frames. In COP/BOP modes, the frame complete status bit is set during the next-to-last bit (on TxD pin) of the last character in the data/information field. In BOP mode, if a 1-bit residual character is selected through OMR[7:5], then this bit is set during the next-to-last bit (on TxD pin) of the last full length character of the information field.

[4] Send Break Ack (ASYNC)/Send SOM ACK (COP)/Send SOM-Abort Ack (BOP) —

ASYNC	Set when the transmitter begins transmission of a break in response to the send break command. If the command is reinvoked, the bit will be set again at the beginning of the next character time. The user can control the length of the break by counting character times through this mechanism.
COP	Set when the transmitter begins transmission of a SYN pattern in response to the TSOM or TSOMP command. If the command is reinvoked, the bit will be set again at the beginning of the next transmitted SYN pattern. The user can control the number of SYNs which are sent through this mechanism.
BOP	Set when the transmitter begins transmission of a FLAG/ABORT in response to the TSOM or TSOMP or TABRK command. If the command is reinvoked, the bit will be set again at the beginning of the next transmitted FLAG/ABORT. The user can control the number of FLAGs/ABORTs which are sent through this mechanism.

[3] DPLL Error — Set while the DPLL is operating in FM mode to indicate that a data transition was not detected within the detection window for two consecutive bits and that the DPLL was forced into search mode. This feature is disabled when the DPLL is specified as the clock source for the transmitter via TTR[6:4].

[2:0] Received Residual Character Length (BOP) —

BOP	This field should be examined to determine the length of the last character of the I field (character tagged with REOM status bit) if RSR[0] is set to indicate that the length was not equal to the character length specified in RPR[1:0]. This field is negated when a reset receiver or disable receiver command is issued, or when the first control character for the next frame of data is in HSRL (see Figure 1). Care must be taken to read TRSR[2:0] before these bits are cleared.

[1] Receiver in Hunt Mode (COP) —

COP	This bit is asserted after the receiver is reset or disabled. It indicates that the receiver is in the hunt mode, searching the data stream for a SYN sequence to establish character synchronization. The bit is negated automatically when character sync is achieved.

[0] Receiver in Transparent Mode (BISYNC) —

COP	Indicates that a DLE-STx sequence was received and the receiver is operating in BISYNC transparent mode. Set two bit times after detection of STx in HSRL. See Figure 1 for receiver data path. Transparent mode operation is terminated and the bit is negated automatically when one of the terminators for transparent text mode is received (DLE-ETx/ETB/ITB/ENQ).

Input and Counter/Timer Status Register (ICTSRA, ICTSRB)

This register informs the CPU of status of the counter/timer and inputs. The logical-OR of bits [6:4] is presented in GSR[3] or GSR[7] for channels A and B, respectively. Unless otherwise specified, bits of this register are reset only:

1. By performing a write operation to the status register with the bits to be reset (ones in the accompanying data word for bits [6:4] only).
2. When the RESETN input is asserted (bits [7:4]) only.

[7] Counter/Timing Running — Set when the C/T is started by start C/T command and reset when it is stopped by a stop C/T command.

[6] Counter/Timer Zero Detect — Set when the counter/timer reaches zero count, or when the bit length measurement is enabled (CTCR [2:0] = 011) and the RXD input has returned high. The assertion of this bit causes an interrupt to be generated if ICTCR[7] and the channel's master interrupt enable (ICR[1] or ICR[0]) are asserted.

[5] Delta DCD — The DCD input is sampled approximately every 6.8μs using the 32×, 4800 baud output from the BRG. After synchronizing with the sampling clock, at least two consecutive samples at the same level are required to establish the level. As a consequence, a change of state at the DCD input, lasting at least 17μs, will set this bit. The reset circuitry initializes the sampling circuits so that a change is not falsely indicated at power on time. The assertion of this bit causes an interrupt to be generated if IER[7] and the channel's master interrupt enable (ICR[1] or ICR[0]) are asserted.

[4] Delta CTS/LC — When not in loop mode, the CTS input is sampled approximately every 6.8μs using the 32×, 4800 baud output from the BRG. After synchronizing with the sampling clock, at least two consecutive samples at the same level are required to establish the level. As a consequence, a change of state at the CTS input, lasting at least 17μs, will set this bit. The reset circuitry initializes the sampling circuits so that a change is not falsely indicated at power on time. The assertion of this bit causes an interrupt to be generated if IER[7] and the channel's master interrupt enable (ICR[1] or ICR[0]) are asserted.

In SDLC loop mode, this bit is set upon transitions of the LC output. LC is asserted in response to the 'go on-loop' command when the receiver detects a zero followed by seven ones, and negated in response to the 'go off-loop' command when the receiver detects a sequence of eight ones.

[3:0] Current State of DCD, CTS, GPI2, and GPI1 Inputs — This field provides the current state of the channel's input pins. The bit's value is latched at the beginning of the read cycle.

Interrupt Vector Register (IVR) and Modified Vector Register (IVRM)

[7:0] Register Content — If ICR[2] = 0, the content of IVR register is output on the data bus when the DUSCC has issued an interrupt request and the responding interrupt acknowledge (IACKN) is received. The value in the IVR is initialized to H'0F' on master reset. If 'vector includes status' is specified by

ICR[2] = 1, bits [2:0] or [4:2] (depending on ICR[3]), of the vector are modified as shown in Table 8 to indicate the highest priority interrupt currently active. The priority is programmable through the ICR. This modified vector is stored in the IVRM. When ICR[2] = 1, the content of the IVRM is output on to the data bus on the interrupt acknowledge. The vector is not modified, regardless of the value of ICR[2], if the CPU has not written an initial vector into this register.

Either the modified or unmodified vector can also be read by the CPU via a normal bus read cycle (see Table 1). The vector value is locked at the beginning of the IACK or read cycle until the cycle is completed.

Interrupt Control Register (ICR)

[7:6] Channel A/B Interrupt Priority — Selects the relative priority between channels A and B. The state of this bit determines the value of the interrupt vector (see Interrupt Vector Register). The priority within each channel, from highest to lowest, is as follows:

0 Receiver ready

1 Transmitter ready

2 Rx/Tx status

3 External or C/T status

00 Channel A has the highest priority. The DUSCC interrupt priorities from highest to lowest are as follows: A(0), A(1), A(2), A(3), B(0), B(1), B(2), B(3)

01 Channel B has the highest priority. The DUSCC interrupt priorities from highest to lowest are as follows: B(0), B(1), B(2), B(3), A(0), A(1), A(2), A(3)

10 Priorities are interleaved between channels, but channel A has the highest priority between events of equal channel priority. The DUSCC interrupt priorities from highest to lowest are as follows: A(0), B(0), A(1), B(1), A(2), B(2), A(3), B(3)

11 Priorities are interleaved between channels, but channel B has the highest priority between events of equal channel priority. The DUSCC interrupt priorities from highest to lowest are as follows: B(0), A(0), B(1), A(1), B(2), A(2), B(3), A(3)

Table 8. Interrupt Status Encoding

IVRM [2:0]/ [4:2]	HIGHEST PRIORITY INTERRUPT CONDITION
000	Channel A receiver ready
001	Channel A transmitter ready
010	Channel A Rx/Tx status
011	Channel A external or C/T status
100	Channel B receiver ready
101	Channel B transmitter ready
110	Channel B Rx/Tx status
111	Channel B external or C/T status

[5:4] Vector Mode — The value of this field determines the response of the DUSCC when the interrupt acknowledge (IACKN) is received from the CPU.

00 or 01 or 10 Vectored mode. Upon interrupt acknowledge, the DUSCC locks its current interrupt status until the end of the acknowledge cycle. If it has an active interrupt pending, it responds with the appropriate vector and then asserts DTACKN. If it does not have an interrupt, it propagates the acknowledge through its X2/IDCN output if this function is programmed in PCRA[7]. Otherwise, the IACKN is ignored. Locking the interrupt status at the leading edge of IACKN prevents a device at a high position in the interrupt daisy chain from responding to an IACK issued for a lower priority device while the acknowledge is being propagated to that device.

11 Non-vectored mode. The DUSCC ignores an IACK if one is received; the interrupt vector is not placed on the data bus. The internal interrupt status is locked when a read of the IVR or IVRM is performed. Except for the absence of the vector on the bus, the DUSCC performs as it does in vectored mode — the vector is prioritized and modified if programmed.

[3] Vector Bits to Modify — Selects which bits of the vector stored in the IVR are to be modified to indicate the highest priority interrupt pending in the DUSCC. See Interrupt Vector Register.

0 Modify bits 2:0 of the vector.

1 Modify bits 4:2 of the vector.

[2] Vector Includes Status — Selects whether the modified (includes status) (IVRM) or unmodified vector (IVR) is output in response to an interrupt acknowledge (see Interrupt Vector Register).

0 Unmodified vector.

1 Modified vector.

[1] Channel A Master Interrupt Enable —

0 Channel A interrupts are disabled.

1 Channel A interrupts are enabled.

[0] Channel B Master Interrupt Enable —

0 Channel B interrupts are disabled.

1 Channel B interrupts are enabled.

General Status Register (GSR)

This register provides a 'quick look' at the overall status of both channels of the DUSCC. A write to this register with 1s at the corresponding bit positions causes TxRDY (bits 5 and 1) and/or RxRDY (bits 4 and 0) to be reset. The other status bits can be reset only by resetting the individual status bits that they point to.

[7] Channel B External or Counter/Timer Status — This bit indicates that one of the following status bits is asserted: ICTSRB[6:4].

[6] Channel B Receiver or Transmitter Status — This bit indicates that one of the following status bits is asserted: RSRB[7:0], TRSRB[7:3].

[5] Channel B Transmitter Ready — The assertion of this bit indicates that one or more characters may be loaded into the channel B transmitter FIFO to be serialized by the transmit shift register. See description of OMR[4]. This bit can be asserted only when the transmitter is enabled. Disabling or resetting the transmitter negates TxRDY.

[4] Channel B Receiver Ready — The assertion of this bit indicates that one or more characters are available in the channel B receiver FIFO to be read by the CPU. See description of OMR[3]. RxRDY is initially reset (negated) by a chip reset or when a 'reset channel B receiver' command is invoked.

[3] Channel A External or Counter/Timer Status — This bit indicates that one of the following status bits is asserted: ICTSRA[6:4].

[2] Channel A Receiver or Transmitter Status — This bit indicates that one of the following status bits is asserted: RSRA[7:0], TRSRA[7:3].

[1] Channel A Transmitter Ready — The assertion of this bit indicates that one or more characters may be loaded into the channel A transmitter FIFO to be serialized by the transmit shift register. See description of OMR[4]. This bit can be asserted only when the transmitter is enabled. Disabling or resetting the transmitter negates TxRDY.

[0] Channel A Receiver Ready — The assertion of this bit indicates that one or more characters are available in the channel A receiver FIFO to be read by the CPU. See description of OMR[3]. RxRDY is initially reset (negated) by a chip reset or when a 'reset channel A receiver' command is invoked.

Channel Command Register (CCRA, CCRB) —

Commands to the DUSCC are entered through the channel command register. The format of that register is shown in Table 9. A

Table 9. Command Register Bit Format

CHANNEL COMMAND REG

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(CCRA, CCRB)	00 = Transmitter CMD		don't care	don't care	**Transmitter Command** 0000 — reset Tx 0001 — reset TxCRC* 0010 — enable Tx 0011 — disable Tx 0100 — transmit SOM (TSOM) 0101 — transmit SOM with PAD (TSOMP) 0110 — transmit EOM (TEOM)* 0111 — transmit ABORT/BREAK (TABRK) 1000 — transmit DLE (TDLE)* 1001 — go active on poll 1010 — reset go active on poll 1011 — go on-loop 1100 — go off-loop 1101 — exclude from CRC*			
	01 = Receiver CMD		don't care	don't care	**Receiver Command** 0000 — reset Rx 0001 — reserved 0010 — enable Rx 0011 — disable Rx			
	10 = C/T CMD		don't care	don't care	**Counter/Timer Command** 0000 — start 0001 — stop 0010 — preset to FFFF 0011 — preset from CTPRH/CTPRL			
	11 = DPLL CMD		don't care	don't care	**DPLL Command** 0000 — enter search mode 0001 — disable DPLL 0010 — set FM mode 0011 — set NRZI mode 0100 — reserved for test 0101 — reserved for test			

NOTE:
*FIFOed commands

read of this register returns the last invoked command (with bits 4 and 5 set to 1).

Transmitter Commands

0000 Reset transmitter. Causes the transmitter to cease operation immediately. The transmit FIFO is cleared and the TxD output goes into the marking state. Also clears the transmitter status bits (TRSR[7:4]) and resets the TxRDY status bit (GSR[1] or GSR[5] for channels A and B, respectively). The counter/timer and other registers are not affected.

0001 Reset transmit CRC. This command is appended to and FIFOed along with the next character loaded into the transmit FIFO. It causes the transmitter CRC generator to be reset to its initial state prior to beginning transmission of the appended character.

0010 Enable transmitter. Enables transmitter operation, conditioned by the state of the CTS ENABLE Tx bit, TPR[2]. Has no effect if invoked when the transmitter has previously been enabled.

0011 Disable transmitter. Terminates transmitter operation and places the TxD output in the marking state at the next occurrence of a transmit FIFO empty condition. All characters currently in the FIFO, or any loaded subsequently prior to attaining an empty condition, will be transmitted.

0100 Transmit start of message. Used in COP and BOP modes to initiate transmission of a frame after the transmitter is first enabled, prior to sending the contents of the FIFO. Can also be used to precisely control the number of SYN/FLAGs at the beginning of transmission or in between frames.

When the transmitter is first enabled, transmission will not begin until this command (or the transmit SOM with PAD command, see below) is issued. The command causes the SYN (COP) or FLAG (BOP) pattern to be transmitted. SEND SOM ACK (TRSR[4]) is set when transmission of the SYN/FLAG begins. The CPU may then reinvoke the command if multiple SYN/FLAGs are to be transmitted. Transmission of the FIFO characters begins when the command is no longer reinvoked. If the FIFO is empty, SYN/FLAGs continue to be transmitted until a character is loaded into the FIFO, but the status bit (TSR[4]) is not set. Insertion of SYN/FLAGs between frames can be accomplished by invoking this command after the frame complete status bit (TRSR[5]) has been asserted in response to transmission of the end-of-message sequence.

0101 Transmit start of message with opening PAD. Used in COP and BOP modes after the transmitter is first enabled to send a bit pattern for DPLL synchronization prior to transmitting the opening SYN (COP) or FLAG (BOP). The SYN/FLAG is sent at the next occurrence of a transmit FIFO empty condition. All characters currently in the FIFO, or any loaded subsequently prior to attaining an empty condition, will be transmitted. While the PAD characters are transmitted, the character length is set to 8 bits, (regardless of the programmed length), and parity generation (COP), zero insertion (BOP), and LRC/CRC accumulation are disabled. SEND SOM ACK (TRSR[4]) is set when transmission of the SYN/FLAG begins. The CPU may then invoke the transmit SOM command if multiple SYN/FLAGs are to be transmitted.

The TSOM/TSOMP commands, described above, are sampled by the controller in alternate bit times of the transmitter clock. As a consequence, the first bit time of a COP/BOP frame will be transmitted on the TxD pin, after a maximum of three bit times, after the command is issued. (The additional 1-bit delay in the data path is due to the data encoding logic.)

0110 Transmit end-of-message. This command is appended to the next character loaded into the transmit FIFO. It causes the transmitter to send the end-of-message sequence (selected FCS in COP modes, FCS – FLAG in BOP modes) after the appended character is transmitted. Frame complete (TRSR[5]) is set when transmission of the FCS begins. This command is also asserted automatically if the TEOM on zero count or done control bit (TPR[4]) is asserted, and the counter/timer is programmed to count transmitted characters when the character which causes the count to go to zero is loaded into the transmit FIFO.

0111 Transmit Abort BOP/Transmit Break ASYNC. In BOP modes, causes an abort (eight ones) to be transmitted after transmission of the character currently in the shift register is completed. The transmitter then sends MARKs or FLAGs depending on the state of underrun control (TPR[7:6]). Send SOM/abort ack (TRSR[4]) is set when the transmission of the abort begins. If the command is reasserted before transmission of the previous ABORT is completed, the process will be repeated. This can be used to send the idle sequence. The 'transmit SOM' command must be used to initiate transmission of a new message. In either mode, invoking this command causes the transmit FIFO to be flushed (characters are not transmitted).

In ASYNC mode, causes a break (space) to be transmitted after transmission of the character currently in the shift register is completed. Send break ack (TRSR[4]) is set when the transmission of the break begins. The transmitter keeps track of character times. If the command is reasserted, send break ack will be set again at the beginning of the next character time. The user can use this mechanism to control the length of the break in character time multiples. Transmission of the break is terminated by issuing a 'reset Tx' or 'disable Tx' command.

1000 Transmit DLE. Used in COP modes only. This command is appended to and FIFOed with the next character loaded into the transmitter FIFO. It causes the transmitter to send a DLE, (EBCDIC H'10', ASCII H'10') prior to transmitting the appended character. If the transmitter is operating in BISYNC transparent mode, the transmitter control logic automatically causes a second DLE to be transmitted whenever a DLE is detected at the top of the FIFO. In this case, the TDLE command should not be invoked. An extra (third) DLE, however, will not be sent if the transmit DLE command is invoked.

1001 Go active on poll. Used in BOP loop mode only. Causes the transmitter, if it is enabled, to begin sending when an EOP sequence consisting of a zero followed by seven ones is detected. The last one of the EOP is changed to zero, making it another FLAG, and then the transmitter operates as described in the detailed operation section. The loop sending status bit (TRSR[6]) is asserted concurrent with the beginning of transmission.

1010 Reset go active on poll. Clears the stored 'go active on poll' command.

1011 Go on-loop. Used in BOP loop mode to control the assertion of the LC_N output. This output provides the means of controlling external loop interface hardware to go on-loop and off-loop. When the command is asserted, the DUSCC will look for the receipt of seven contiguous ones, at which time it will assert the LC_N output and set the delta DCD/LC status bit (ICTSR[4]). This allows the DUSCC to break into the loop without affecting loop operation. This command must be used to initiate loop mode operation.

1100 Go off-loop. Used in BOP loop mode to control the negation of the LC_N output. This output provides the means of controlling external loop interface hardware to go on-loop and off-loop. When the command is asserted, the DUSCC will look for the receipt of eight contiguous ones, at which time it will negate the LC_N output and set the delta DCD/LC status bit (ICTSR[4]). This allows the DUSCC to get off the loop without affecting loop operation. This command is normally used to terminate loop mode operation.

1101 Exclude from CRC. This command is appended to and FIFOed along with the next character loaded into the transmit FIFO. It causes the transmitter CRC generator to be disabled while the appended character is being transmitted. Thus, that character is not included in the CRC accumulation.

Receiver Commands

0000 Reset Receiver. Causes the receiver to cease operation, clears the receiver FIFO, and clears the receiver status (RSR[7:0], TRSR[3:0], and either GSR[0] or GSR[4] for channels A and B, respectively). The counter/timer and other registers are not affected.

0001 Reserved.

0010 Enable receiver. Causes receiver operation to begin, conditioned by the state of the DCD ENABLE Rx bit, RPR[2]. Receiver goes into START, SYN, or FLAG search mode depending on channel protocol mode. Has no effect if invoked when the receiver has previously been enabled.

0011 Disable receiver. Terminates operation of the receiver. Any character currently being assembled will be lost. Does not affect FIFO or any status.

Counter/Timer Commands

0000 Start. Starts the counter/timer and prescaler.

0001 Stop. Stops the counter/timer and prescaler. Since the command may be asynchronous with the selected clock source, the counter/timer and/

or prescaler may count one or more additional cycles before stopping.

0010	Preset to FFFF. Presets the counter timer to H'FFFF' and the prescaler to its initial value. This command causes the C/T output to go low.
0011	Preset from CTPRH/CTPRL. Transfers the current value in the counter/timer preset registers to the counter/timer and presets the prescaler to its initial value. This command causes the C/T output to go low.

Digital Phase-Locked Loop Commands

0000	Enter Search Mode. This command causes the DPLL counter to be set to the value 16 and the clock output will be forced high. The counter will be disabled until a transition on the data line is detected, at which point it will start incrementing and the clock output will go from high to low. After the counter reaches a count of 31, it will reset to zero and cause the clock output to go from low to high. The DPLL will then continue normal operation. This allows the DPLL to be locked onto the data without preframe transitions. This command should not be used if the DPLL is programmed to supply the clock for the transmitter and the transmitter is active.
0001	Disable DPLL. Disables operation of the DPLL.
0010	Set FM Mode. Sets the DPLL to the FM mode of operation, used when FM0, FM1, or Manchester (NRZ) is selected by CMR1[7:6].
0011	Set NRZI Mode. Sets the DPLL to the NRZI mode of operation, used when NRZ or NRZI is selected by CMR1[7:6].
0100	Reserved for test
0101	Reserved for test

DETAILED OPERATION

Interrupt Control

A single interrupt output (IRQN) is provided which is activated upon the occurrence of any of the following conditions:

Channel A external or C/T special condition

Channel B external or C/T special condition

Channel A Rx/Tx error or special condition

Channel B Rx/Tx error or special condition

Channel A TxRDY

Channel B TxRDY

Channel A RxRDY

Channel B RxRDY

Each of the above conditions occupies a bit in the general status register (GSR). If ICR[2] is set, the eight conditions are encoded into three bits which are inserted into bits [2:0] or [4:2] of the interrupt vector register. This forms the content of the IVRM during an interrupt acknowledge cycle. Unmodified and modified vectors can be read directly through specified registers. Two of the conditions are the inclusive OR of several other maskable conditions:

- Ext or C/T special condition: Delta DCD, delta CTS or C/T zero count (ICTSR[6:4]).
- Rx/Tx error or special condition: Any condition in the receiver status register (RSR[7:0]) or a transmitter or DPLL condition in the transmitter and receiver status register (TRSR[7:3]).

The TxRDY and RxRDY conditions are defined by OMR[4] and OMR[3], respectively. Also associated with the interrupt system are the interrupt enable register (IER), one bit in the counter/timer control register (CTCR), and the interrupt control register (ICR).

The IER is programmed to enable specified conditions or groups of conditions to cause an interrupt by asserting the corresponding bit. A negated bit prevents an interrupt from occurring when the condition is active and hence masks the interrupt. In addition to the IER, CTCR[7] could be programmed to enable or disable an interrupt upon the C/T zero count condition. The interrupt priorities within a channel are fixed. Priority between channels is controlled by ICR[7:6]. Refer to Table 8 and ICR[7:6].

The ICR contains the master interrupt enables for each channel (ICR[1] and ICR[0]) which must be set if the corresponding channel is to cause an interrupt. The CPU vector mode is specified by ICR[5:4] which selects either vectored or non-vectored operation. If vectored mode is selected, the content of the IVR or IVRM is placed on the data bus when IACK is activated. If ICR[2] is set, the content of IVRM is output which contains the content of IVR and the encoded status of the interrupting condition.

Upon receiving an interrupt acknowledge, the DUSCC locks its current interrupt status until the end of the acknowledge cycle. If it has an active interrupt pending, it responds with the appropriate vector and then asserts DTACKN. If it does not have an interrupt, it propagates the acknowledge through its X2/IDCN output if this function is programmed in PCRA[7]; otherwise, the IACKN is ignored. Locking the interrupt status at the leading edge of IACKN prevents a device at a high position in the interrupt daisy chain from responding to an IACK issued for a lower priority device while the acknowledge is being propagated to that device.

DMA Control

The DMA control section provides the interface to allow the DUSCC to operate with an external DMA controller. One of four modes of DMA can be programmed for each channel independently via CMR2[5:3]:

- Half-duplex single address. In this mode, a single pin provides both DMA read and write requests. Acknowledgement of the requests is via a single DMA acknowledge pin. The data transfer is accomplished in a single bus cycle — the DMA controller places the memory address of the source or destination of the data on the address bus and then issues the acknowledge signal, which causes the DUSCC to either write the data into its transmit FIFO (write request) or to output the contents of the top of the receive FIFO (read request). The cycle is completed when the DTCN input is asserted by the DMA controller. This mode can be used when channel operation is half-duplex (e.g., BISYNC). It allows a single DMA channel to service the receiver and transmitter.
- Half-duplex dual address. In this mode, a single pin provides both DMA read and write requests. Acknowledgement of the requests is via normal bus read and write cycles. The data transfer requires two bus cycles — the DMA controller acquires the data from the source (memory for a Tx DMA or DUSCC for a Rx DMA) on the first cycle and deposits it at the destination (DUSCC for a Tx DMA or memory for a Rx DMA) on the second bus cycle. This mode is used when channel operation is half-duplex (e.g., BISYNC) and allows a single DMA channel to service the receiver and transmitter.
- Full-duplex single address. This mode is similar to half-duplex single address mode but provides separate request and acknowledge pins for the receiver and transmitter.
- Full-duplex dual address. This mode is similar to half-duplex dual address mode but provides duplex dual address mode but provides separate request pins for the receiver and transmitter.

Figures 4 through 7 describe operation of the DUSCC in the various DMA environments. Table 10 summarizes pins used for the DMA request and acknowledge function for the transmitter and receiver for the different DMA modes.

Table 10. DMA REQ and ACK Pins for Operational Modes

FUNCTION	HALF DUPLEX SINGLE ADDR DMA	HALF DUPLEX DUAL ADDR DMA	FULL DUPLEX SINGLE ADDR DMA	FULL DUPLEX DUAL ADDR DMA
RCVR REQ	RTxDRQ_N	RTxDRQ_N	RTxDRQ_N	RTxDRQ_N
TRAN REQ	Same as RCVR REQ	Same as RCVR REQ	TxDRQ_N	TxDRQ_N
RCVR ACK	RTxDAK_N	Normal read RCVR FIFO	RTxDAK_N	Normal read RCVR FIFO
TRAN ACK	Same as RCVR ACK	Normal write TRAN FIFO	TxDAK_N	Normal write TRAN FIFO

The DMA request signals are functionally identical to the TxRDY and RxRDY status signals for each serial channel except that in DMA the signals are negated on the leading edge of the acknowledge signal when the subsequent transfer causes the FIFO to become full (transmitter request) or empty (receiver request). In non DMA operation TxRDY and RxRDY signals are negated only after the transfer is completed. The DMA read request can be programmed through OMR[3] be asserted either when any character is in the receive FIFO or only when the receive FIFO is full. Likewise, the DMA write request can be programmed through OMR[4] to be asserted either when the transmit FIFO is not full or only when the transmit FIFO is empty (the transmitter must be enabled for a DMA request to be asserted). The request signals are negated when the respective data transfer cycle is completed. When the serial channel is not operating in DMA mode, the request and acknowledge pins for the channel can be programmed for other functions (see pin descriptions).

DMA DONEN Operation

As an input, DONEN is asserted by the DMA controller concurrent with the corresponding DMA acknowledge to indicate to the DUSCC that the character being transferred into the TxFIFO is the last character of the transmission frame. In synchronous modes, the DUSCC can be programmed through TPR[4] to automatically transmit the frame termination sequence (e.g., FCS – FLAG in BOP mode) upon receipt of this signal.

As an output, DONEN is asserted by the DUSCC under the following conditions:

a. In response to the DMA acknowledge for a receiver DMA request if the FIFOed RECEIVED EOM status bit (RSR[7]) is set for the character being transferred.

b. In response to the DMA acknowledge for a transmitter DMA request if the counter/timer has been programmed to count transmitted characters and the terminal count has occurred.

Block Transfers Using DTACK

The DTACKN line may be used to synchronize data transfers to and from the DUSCC utilizing a 'wait' state. Either the receiver or the transmitter or both may be programmed for this mode of operation, independently for each channel, via CMR2[5:3].

In this mode, if the CPU attempts a write to the transmit FIFO and an empty FIFO position is not available, the DTACK line will remain negated until a position empties. The data will then be written into the FIFO and DTACKN will be asserted to signify that the transfer is complete.

Similarly, a read of an empty receive FIFO will be held off until data is available to be transferred. Potentially, this mode can cause the microcomputer system to hang up if, for example, a read request was made and no further data was available.

Timing Circuits

The timing block for each channel consists of a crystal oscillator, a bit rate generator (BRG), a digital phase locked loop (DPLL) and a 16-bit counter/timer (C/T) (see Figure 8).

Crystal Oscillator

The crystal oscillator operates directly from a crystal (normally 14.7456MHz if the internal BRG is to be used) connected across the X1/CLK and X2/IDCN pins with a minimum of external components. If an external clock of the appropriate frequency is available, it may be connected to the X1/CLK pin. This signal is divided by two to provide the internal system clock (a maximum of 16MHz input is allowed).

Bit Rate Generator

The BRG operates from the oscillator or external clock and is capable of generating 16 bit rates. These are available to the receiver, transmitter, DPLL, and C/T. The BRG output is at 32× the base bit rate. Since all sixteen rates are generated simultaneously, each receiver and transmitter may select its bit rate independently. The transmitter and receiver timing registers include a 4-bit field for this purpose (TTR[3:0], RTR[3:0]).

Digital Phase-Locked Loop

Each channel of the DUSCC includes a DPLL used in synchronous modes to recover clock information from a received data stream. The DPLL is driven by a clock at nominally 32 times the data rate. This clock can be programmed, via RTR (7:4), to be supplied from an external input, from the receiver BRG, from the C/T, or directly from the crystal oscillator.

The DPLL uses this clock, along with the data stream to construct a data clock which may then be used as the DUSCC receive clock, transmit clock, or both. The output of the DPLL is a square wave at 1× the data rate. The derived clock can also be programmed to be output on a DUSCC pin; only the DPLL receiver output clock is available at the TRxC pin. Four commands are associated with DPLL operation: Enter search mode, set FM mode, set NRZI mode, and disable DPLL. The commands are described in the command register description. Waveforms associated with the DPLL are illustrated in Figure 9.

DPLL NRZI Mode Operation

This mode is used with NRZ and NRZI data encoding. With this type of encoding, the transitions of the data stream occur at the beginning of the bit cell. The DPLL has a six-bit counter which is incremented by a 32× clock. The first edge detected during search mode sets the counter to 16 and begins operation. The DPLL output clock then rises at a count of 0 and falls at 16. Data is sampled on the rising edge of the clock. When a transition in the data stream is detected, the count length is adjusted by one or two counts, depending on the counter value when the transition occurs (see Table 11). A transition detection at the roll-over point (third column in Figure 11) is treated as a transition occurring at zero count.

The count length adjustments cause the rising edge of the DPLL output clock to converge to the nominal center of the bit cell. In the worst case, which occurs when a DPLL pulse is coincident with the data edge, the DPLL converges after 12 data transitions.

For NRZ encoded data, a stream of alternating ones and zeros should be used as a synchronizing pattern. For NRZI encoded data, a stream of zeros should be used.

Table 11. NRZI Mode Count Length

COUNT WHEN TRANSITION DETECTED	COUNT LENGTH ADJUST-MENT	COUNTER RESET AFTER COUNT REACHES
0 – 7	−2	29
8 – 15	−1	30
16 – 23	+1	32
24 – 30	+2	33
None detected	0	31

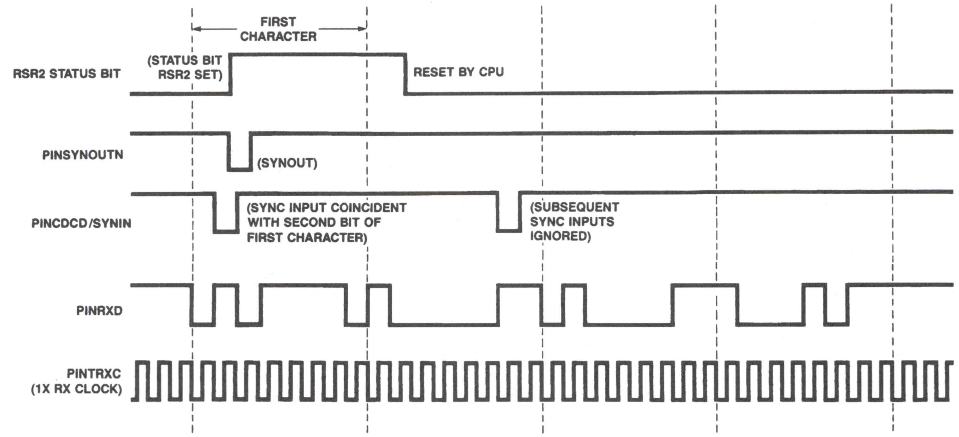


Figure 2. External Sync Mode

DPLL FM Mode Operation

FM operation is used with FM0, FM1, and Manchester data encoding. With this type of encoding, transitions in the data stream always occur at the beginning of the bit cell for FM0 and FM1, or at the center of the bit cell for Manchester. The DPLL 6-bit counter is incremented by a 32× clock. The first edge detected during search mode sets the counter to 16 and begins operation. The DPLL receiver clock then rises on a count of 8 and falls on 24. (The DPLL transmitter clock output falls on a count of 16. It rises on a count of 0 if a transition has been detected between counts of 16 and 23. For other cases, it rises 1/2 count of the 32× input clock sooner.) This provides a 1× clock with edges positioned at the nominal centers of the two halves of the bit cell. The transition detection circuit is enabled between counts of 8 and 23, inclusive. When a transition is detected, the count length is adjusted by one, depending on when the transition occurs (see Table 12).

Table 12. FM Mode Count Length

COUNT WHEN TRANSITION DETECTED	COUNT LENGTH ADJUST-MENT	COUNTER RESET AFTER COUNT REACHES
8 – 15	–1	30
16 – 23	+1	32
24 – 7	Disabled	
None detected	0	31

If a transition is not detected for two consecutive data bits, the DPLL is forced into search mode and the DPLL error status bit (TRSR [3]) is asserted. This feature is disabled when the DPLL output is used only as the transmitter clock.

To prevent the DPLL from locking on the wrong edges of the data stream, an opening PAD sequence should be transmitted. For FM0, a stream of at least 16 ones should be sent initially. For FM1, a minimum stream of 16 zeros should be sent and for Manchester encoding the initial data stream should consist of alternating ones and zeros.

Counter/Timer

Each channel of the DUSCC contains a counter/timer (C/T) consisting of a 16-bit down counter, a 16-bit preset register, and associated control circuits. Operation of the counter/timer is programmed via the counter/timer control register (CTCR). There are also four commands associated with C/T operation, as described in the Command Description section.

The C/T clock source, clock prescaling, and operating mode are programmed via CTCR[2:0], CTCR[4:3], and CTCR[6] respectively. The preset register is loaded with a minimum of 2 by the CPU and its contents can be transferred into the down counter by a command, or automatically upon reaching terminal count if CTCR[6] is negated. Commands are also available to stop and start the C/T and to preset it to an initial value of FFFF. Counting is triggered by the falling edge of the clocking input. The C/T zero count status bit, ICTSR[6], is set when the C/T reaches the terminal count of zero and ICTSR[7] indicates whether the counter is currently enabled or not.

An interrupt is generated upon reaching zero count if CTCR[7] and the channel's master interrupt enable are asserted. The output of the C/T can be programmed to be output on the channel's RTxC or TRxC pin (via PCR[4:0]) as either a single pulse or a square wave, as programmed in CTCR[5]. The contents of the C/T can be read at any time by the CPU, but the C/T should normally be stopped before this is done. Several C/T operating modes can be selected by programming of the counter/timer control register. Typical applications include:

1. Programmable divider. The selected clock source, optionally prescaled, is divided by the contents of the preset register. The counter automatically reloads itself each time the terminal count is reached. In this mode, the C/T may be programmed to be used as the Rx or Tx bit rate generator, as the input to the DPLL, or it may be output on a pin as either a pulse or a square wave. The C/T interrupt should be disabled in this mode.
2. Periodic interrupt generator. This mode is similar to the programmable divider mode, except that the C/T interrupt is enabled, resulting in a periodic interrupt to the CPU.
3. Delay timer. The counter is preset from the preset register and a clock source, optionally prescaled, is selected. An interrupt is generated upon reaching terminal count. The C/T continues counting

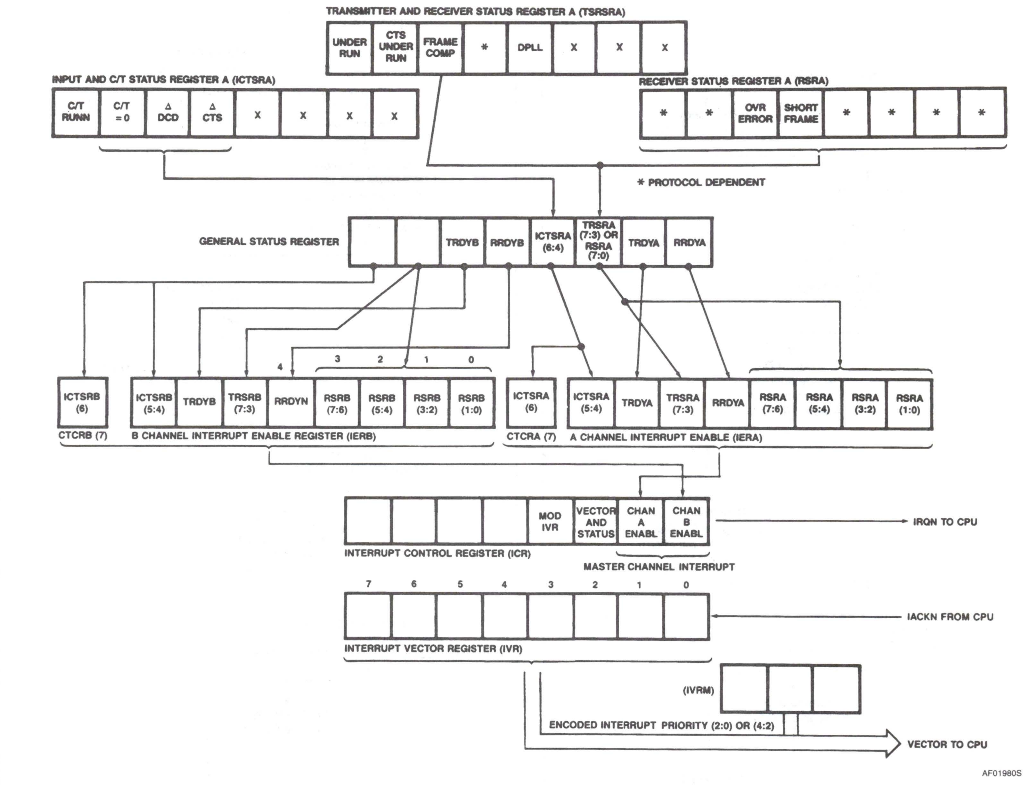

TRANSMITTER AND RECEIVER STATUS REGISTER A (TSRSRA)
UNDER RUN
CTS UNDER RUN
FRAME COMP
*
DPLL
X
X
X
INPUT AND C/T STATUS REGISTER A (ICTSRA)
C/T RUNN
C/T = 0
Δ DCD
Δ CTS
X
X
X
X
RECEIVER STATUS REGISTER A (RSRA)
*
*
OVR ERROR
SHORT FRAME
*
*
*
*
* PROTOCOL DEPENDENT
GENERAL STATUS REGISTER
TRDYB
RRDYB
ICTSRA (6:4)
TRSRA (7:3) OR RSRA (7:0)
TRDYA
RRDYA
4
3
2
1
0
ICTSRB (6)
ICTSRB (5:4)
TRDYB
TRSRB (7:3)
RRDYN
RSRB (7:6)
RSRB (5:4)
RSRB (3:2)
RSRB (1:0)
ICTSRA (6)
ICTSRA (5:4)
TRDYA
TRSRA (7:3)
RRDYA
RSRA (7:6)
RSRA (5:4)
RSRA (3:2)
RSRA (1:0)
CTCRB (7)
B CHANNEL INTERRUPT ENABLE REGISTER (IERB)
CTCRA (7)
A CHANNEL INTERRUPT ENABLE (IERA)
MOD IVR
VECTOR AND STATUS
CHAN A ENABL
CHAN B ENABL
IRQN TO CPU
INTERRUPT CONTROL REGISTER (ICR)
MASTER CHANNEL INTERRUPT
7
6
5
4
3
2
1
0
IACKN FROM CPU
INTERRUPT VECTOR REGISTER (IVR)
(IVRM)
ENCODED INTERRUPT PRIORITY (2:0) OR (4:2)
VECTOR TO CPU
AF01980S

Figure 3. Interrupt Control and Status Register

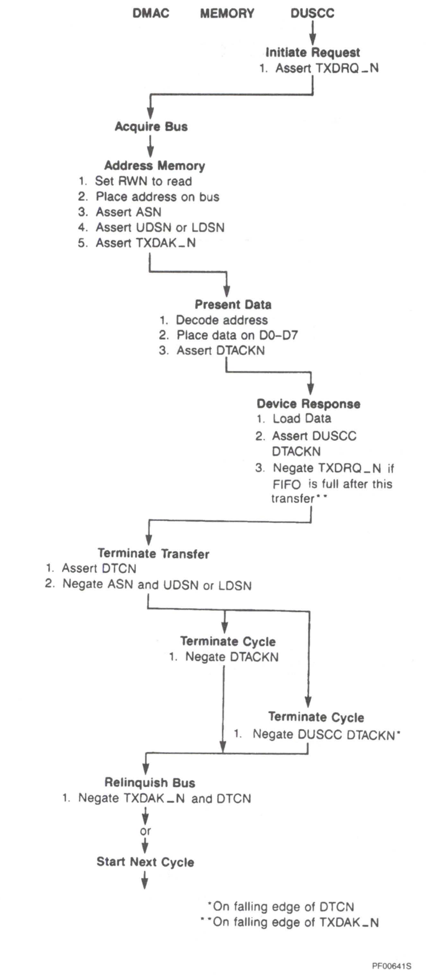
without reloading itself and its contents may be read by the CPU to allow additional delay past the zero count to be determined.

4. Character counter. The counter is preset to FFFF by command and the clock source becomes the internal signal used to control loading of the Rx or Tx characters. This operation is selected by CTCR_ [2:0]. The C/T counts characters loaded into the RxFIFO by the receiver or loaded into the transmit FIFO by the CPU respectively. The current character count can be determined by the CPU by reading the contents of the C/T and taking its ones complement. Optionally, a preset number may be loaded into the counter and an interrupt generated when the count is exhausted. When counting Tx characters, the terminal count condition can be programmed through TPR_ [4] to cause an end of message sequence to be transmitted. When counting received characters, the FIFOed EOM status bit is asserted when the character which causes the count to go to zero is loaded into the receive FIFO. The channel's 'reset Tx' or 'reset Rx' commands have no effect on the operation of the C/T.
5. External event counter. The counter is preset to FFFF by command and an external clock source is selected. The current count can be determined by the CPU by reading the contents of the C/T and taking its ones complement. Optionally, a preset number may be loaded into the counter and an interrupt generated when the count is exhausted.
6. Bit length measurement. The counter is preset to FFFF by command and the X1/CLK/4 clock input gated by RxD mode (optionally prescaled) is programmed. The C/T starts counting when RxD goes low and stops counting when RxD goes high. At this time, ICTSR[6] = 1 is set and an interrupt (if enabled) is generated. The resulting count in the counter can be read by the CPU to determine the bit rate of the input data. Normally this function is used for asynchronous operation.

Communication Channels A and B

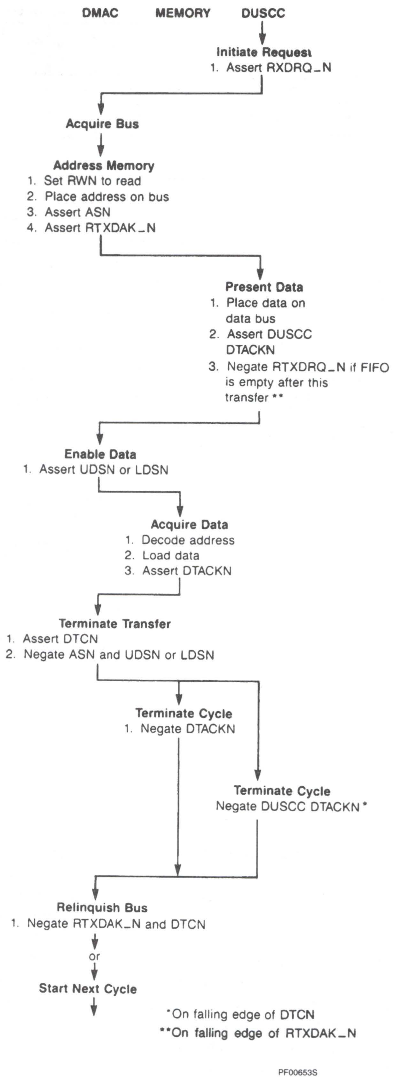
Each communication channel of the DUSCC is a full-duplex receiver and transmitter that supports ASYNC, COP, and BOP transmission formats. The bit rate clock for each receiver and transmitter can be selected independently to come from the bit rate generator, C/T, DPLL, or an external input (such as a modem generated clock).

Dual Universal Serial Communications Controller (DUSCC) SCN68562


DMAC
MEMORY
DUSCC
Initiate Request
1. Assert TXDRQ_N
Acquire Bus
Address Memory
1. Set RWN to read
2. Place address on bus
3. Assert ASN
4. Assert UDSN or LDSN
5. Assert TXDAK_N
Present Data
1. Decode address
2. Place data on D0–D7
3. Assert DTACKN
Device Response
1. Load Data
2. Assert DUSCC DTACKN
3. Negate TXDRQ_N if FIFO is full after this transfer**
Terminate Transfer
1. Assert DTCN
2. Negate ASN and UDSN or LDSN
Terminate Cycle
1. Negate DTACKN
Terminate Cycle
1. Negate DUSCC DTACKN*
Relinquish Bus
1. Negate TXDAK_N and DTCN
or
Start Next Cycle
PF00641S

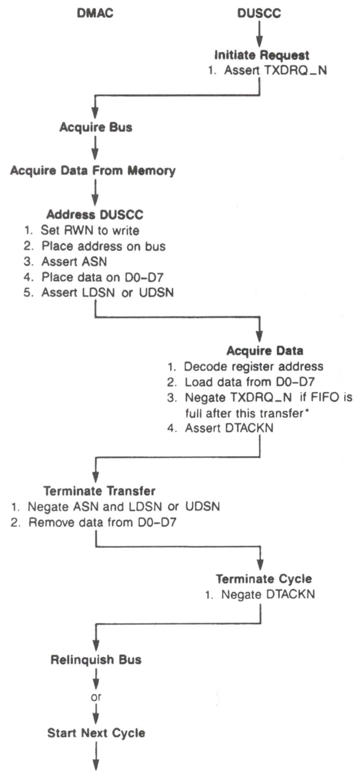
*On falling edge of DTCN
**On falling edge of TXDAK_N

Figure 4. Transmitter DMA Request Operation — Single Address Mode


DMAC
MEMORY
DUSCC
Initiate Request
1. Assert RXDRQ_N
Acquire Bus
Address Memory
1. Set RWN to read
2. Place address on bus
3. Assert ASN
4. Assert RTXDAK_N
Present Data
1. Place data on data bus
2. Assert DUSCC DTACKN
3. Negate RTXDRQ_N if FIFO is empty after this transfer **
Enable Data
1. Assert UDSN or LDSN
Acquire Data
1. Decode address
2. Load data
3. Assert DTACKN
Terminate Transfer
1. Assert DTCN
2. Negate ASN and UDSN or LDSN
Terminate Cycle
1. Negate DTACKN
Terminate Cycle
Negate DUSCC DTACKN *
Relinquish Bus
1. Negate RTXDAK_N and DTCN
or
Start Next Cycle
PF00653S

*On falling edge of DTCN
**On falling edge of RTXDAK_N

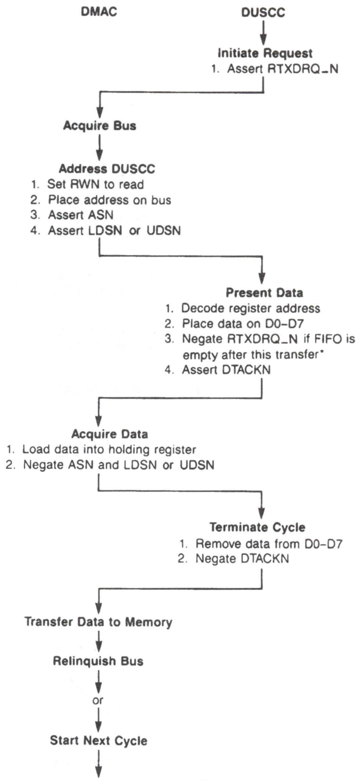
Figure 5. Receiver DMA Request Operation — Single Address Mode


DMAC
DUSCC
Initiate Request
1. Assert TXDRQ_N
Acquire Bus
Acquire Data From Memory
Address DUSCC
1. Set RWN to write
2. Place address on bus
3. Assert ASN
4. Place data on D0–D7
5. Assert LDSN or UDSN
Acquire Data
1. Decode register address
2. Load data from D0–D7
3. Negate TXDRQ_N if FIFO is full after this transfer*
4. Assert DTACKN
Terminate Transfer
1. Negate ASN and LDSN or UDSN
2. Remove data from D0–D7
Terminate Cycle
1. Negate DTACKN
Relinquish Bus
or
Start Next Cycle

*On falling edge of CSN

PF00661S

Figure 6. Transmitter DMA Request Operation — Dual Address Mode

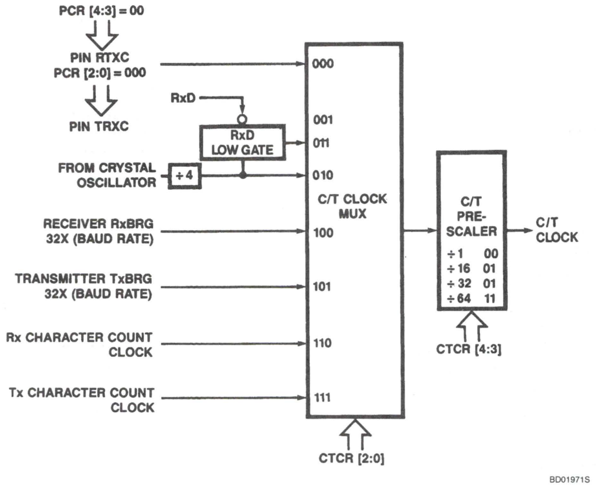

DMAC
DUSCC
Initiate Request
1. Assert RTXDRQ_N
Acquire Bus
Address DUSCC
1. Set RWN to read
2. Place address on bus
3. Assert ASN
4. Assert LDSN or UDSN
Present Data
1. Decode register address
2. Place data on D0–D7
3. Negate RTXDRQ_N if FIFO is empty after this transfer*
4. Assert DTACKN
Acquire Data
1. Load data into holding register
2. Negate ASN and LDSN or UDSN
Terminate Cycle
1. Remove data from D0–D7
2. Negate DTACKN
Transfer Data to Memory
Relinquish Bus
or
Start Next Cycle

*On falling edge of CSN

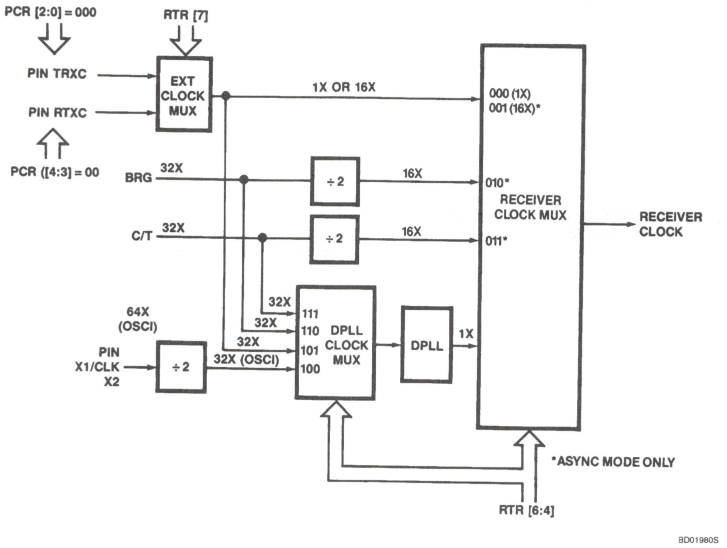
PF00671S

Figure 7. Receiver DMA Request Operation — Dual Address Mode

COUNTER/TIMER
CLOCK SOURCE

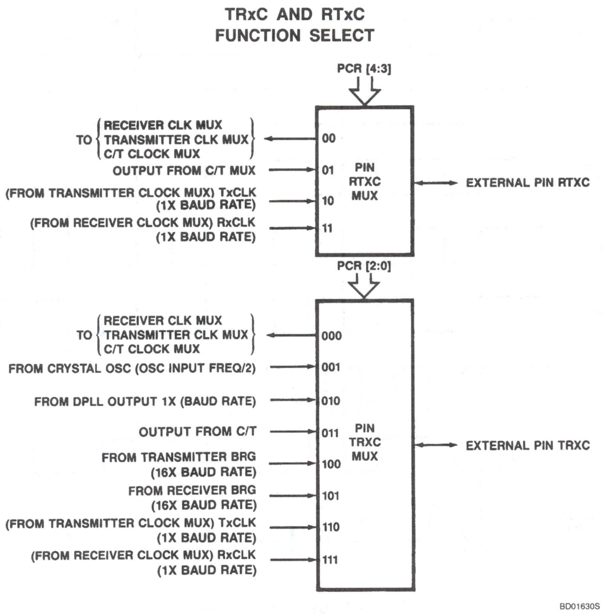

PCR [4:3] = 00
PIN RTXC
PCR [2:0] = 000
PIN TRXC
RxD
RxD LOW GATE
FROM CRYSTAL OSCILLATOR
÷4
RECEIVER RxBRG 32X (BAUD RATE)
TRANSMITTER TxBRG 32X (BAUD RATE)
Rx CHARACTER COUNT CLOCK
Tx CHARACTER COUNT CLOCK
000
001
011
010
C/T CLOCK MUX
100
101
110
111
CTCR [2:0]
C/T PRE-SCALER
÷1 00
÷16 01
÷32 01
÷64 11
CTCR [4:3]
C/T CLOCK
BD01971S

Receiver Clock
Sources


PCR [2:0] = 000
PIN TRXC
PIN RTXC
PCR ([4:3] = 00
RTR [7]
EXT CLOCK MUX
1X OR 16X
BRG
32X
C/T
32X
÷2
16X
÷2
16X
64X (OSCI)
PIN X1/CLK X2
÷2
32X (OSCI)
32X
32X
32X
111
110
101
100
DPLL CLOCK MUX
DPLL
1X
000 (1X)
001 (16X)*
010*
RECEIVER CLOCK MUX
011*
RECEIVER CLOCK
*ASYNC MODE ONLY
RTR [6:4]
BD01980S

NOTE:
OSCI = crystal or EXT generator input

Figure 8. Timing Block


TRxC AND RTxC
FUNCTION SELECT
PCR [4:3]
TO
RECEIVER CLK MUX
TRANSMITTER CLK MUX
C/T CLOCK MUX
OUTPUT FROM C/T MUX
(FROM TRANSMITTER CLOCK MUX) TxCLK (1X BAUD RATE)
(FROM RECEIVER CLOCK MUX) RxCLK (1X BAUD RATE)
00
01
10
11
PIN RTXC MUX
EXTERNAL PIN RTXC
PCR [2:0]
TO
RECEIVER CLK MUX
TRANSMITTER CLK MUX
C/T CLOCK MUX
FROM CRYSTAL OSC (OSC INPUT FREQ/2)
FROM DPLL OUTPUT 1X (BAUD RATE)
OUTPUT FROM C/T
FROM TRANSMITTER BRG (16X BAUD RATE)
FROM RECEIVER BRG (16X BAUD RATE)
(FROM TRANSMITTER CLOCK MUX) TxCLK (1X BAUD RATE)
(FROM RECEIVER CLOCK MUX) RxCLK (1X BAUD RATE)
000
001
010
011
100
101
110
111
PIN TRXC MUX
EXTERNAL PIN TRXC
BD01630S

NOTE:
OSCI = crystal or EXT generator input

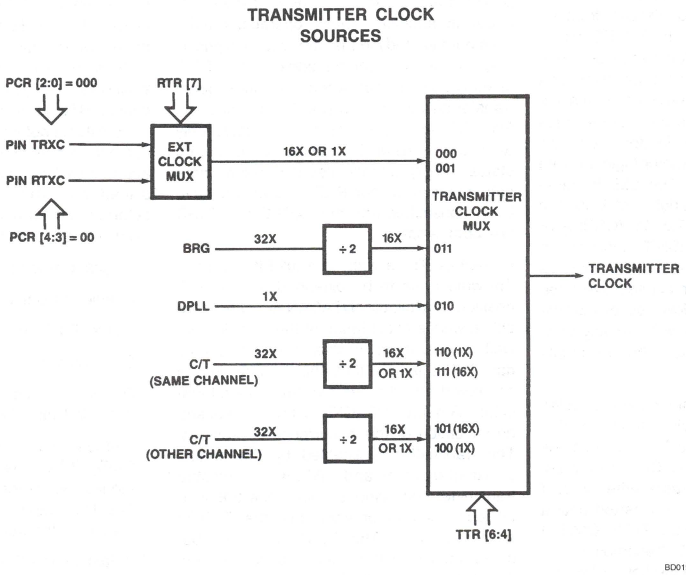

TRANSMITTER CLOCK
SOURCES
PCR [2:0] = 000
RTR [7]
PIN TRXC
PIN RTXC
PCR [4:3] = 00
EXT CLOCK MUX
16X OR 1X
000
001
TRANSMITTER CLOCK MUX
BRG
32X
÷ 2
16X
011
DPLL
1X
010
C/T (SAME CHANNEL)
32X
÷ 2
16X OR 1X
110 (1X)
111 (16X)
C/T (OTHER CHANNEL)
32X
÷ 2
16X OR 1X
101 (16X)
100 (1X)
TRANSMITTER CLOCK
TTR [6:4]
BD01990S

Figure 8. Timing Block (Continued)

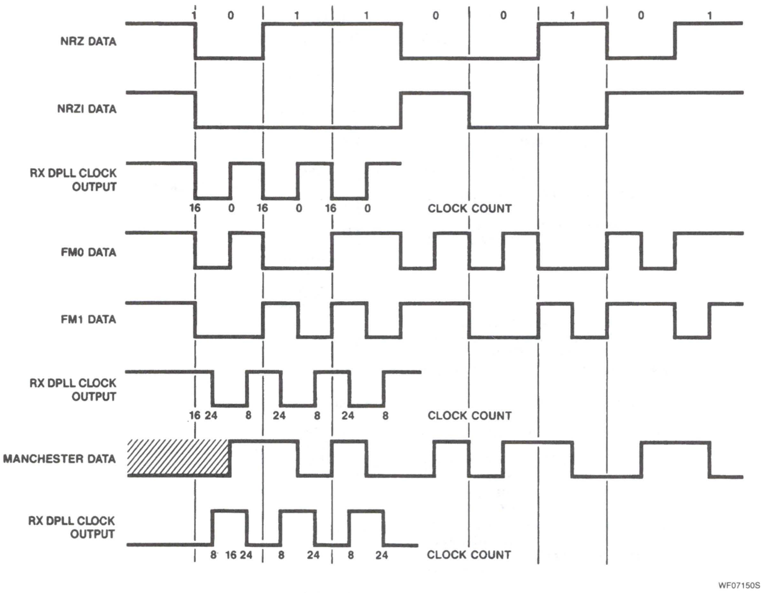


Figure 9. DPLL Waveforms

TRANSMITTER

Transmitter TxFIFO and TxRDY

The transmitter accepts parallel data from the data bus and loads it into the TxFIFO, which consists of four 8-bit holding registers. This data is then moved to the transmitter shift register, TxSR, which serializes the data according to the transmission format programmed. The TxSR is loaded from the TxFIFO, from special character logic, or from the CRC/LRC generator. The LSB is transmitted first, which requires right justification of characters by the CPU. TxRDY (GSR[5] or GSR[1]) and underrun (TRSR[7]) indicate the state of the TxFIFO. The TxFIFO may be addressed at any of four consecutive locations (see Table 1) to allow use of multiple byte word instructions. A write to any valid address always writes data to the next empty FIFO location.

TxRDY is set when the transmitter is enabled and there is an empty position in the TxFIFO (OMR[4] = 0) or when the TxFIFO becomes empty (OMR[4] = 1). The CPU may reset TxRDY through a status reset write cycle. If this is done, it will not be reasserted until a character is transferred to the TxSR (OMR[4] = 0) or when the TxFIFO becomes empty again (OMR[4] = 1). The assertion of TxRDY, enabling of the IER [6] and the enabling of the channel master interrupt ICR [0] or [1] allow an interrupt to be generated.

If DMA operation is programmed, either RTxDRN (half-duplex) or TxDRQN (full-duplex) follows the state of TxRDY if the transmitter is enabled. These operations differ from normal ready in that the request signal is negated on the leading edge of the DMA acknowledge signal when the subsequent transfer causes the transmit FIFO to become full, while the TxRDY signal is negated only after the transfer is completed. Underrun status TRS[7] set indicates that one or more data characters (not PAD characters) have been transmitted and the TxFIFO and TxSR are both empty.

In 'wait on Tx', a write to a full FIFO causes the write cycle to be extended until a FIFO position is available. DTACKN is asserted to acknowledge acceptance of the data. In non-wait modes, if an attempt is made to load data into a full TxFIFO, the TxFIFO data is preserved and the overrun data character(s) is lost. A normal DTACKN will be issued, and no indication of this occurrence is provided. The transmitter is enabled by the enable transmitter command. When the disable transmitter command is issued, the transmitter continues to operate until the TxFIFO becomes empty. The TxRDY does not become valid until the transmitter is enabled. Characters can be loaded into the FIFO while disabled. However, if the FIFO is full when the transmitter is enabled, TxRDY is not asserted.

TxRTS Control

If TxRTS CONTROL, TPR[3], is programmed, the channel's RTS output is negated five bit times after the last bit (stop bit in ASYNC mode) of the last character is transmitted. RTS is normally asserted and negated by writing to OMR_[0]. The assertion of TPR[3] causes RTS to be reset automatically (if the transmitter is not enabled) after all characters in the transmitter FIFO (if any) are transmitted and five bit times after the 'last character' is shifted out. This feature can be used to automatically terminate the transmission of a message as follows:

- Program auto-reset mode: TPR_[3] = 1.
- Enable transmitter.
- Assert RTS_N: OMR_[0] = 1.
- Send message.
- Disable transmitter after the last character is loaded into the TxFIFO.
- The last character will be transmitted and OMR_ [0] will be reset five bit times after the last bit, causing RTS_N to be negated. The TxD output will remain in the marking state until the transmitter is enabled again.

The 'last bit' in ASYNC is simply the last stop bit of the character. In BOP and COP, the last character is defined either explicitly by either appending it with TEOM or implicitly through the selection of the frame underrun control sequence, TPR[7:6] (transmitter parameter register). Table 13 summarizes the relationship of the selected underrun sequence and the protocol mode.

Tx CTS Operation
If CTS enable Tx, TPR[2], is set, the CTSN input must be asserted for the transmitter to operate. Changes in CTSN while a character is being transmitted do not affect transmission of that character. However, if the CTS input becomes negated when TPR[2] is set and the transmitter is enabled and ready to start sending a new character, CTS underrun, TRSR[6], is asserted and the TxD output is placed in the marking (high) state. In ASYNC mode, operation resumes when CTSN is asserted again. In COP and BOP modes, the transmission of the message is terminated and operation of the transmitter will not resume until CTS is asserted and a TSOM or TSOMP command is invoked. Prior to issuing the command and retransmitting the message, the transmitter must be reset. After a change-of-state CTS is established by the input sampling circuits (refer to the description of ICTSR[4]), it is sampled by the Tx controller 1½ bit times before each new character is serialized out of the Tx shift register. (This is 2½ bits before the LSB of the new character appears on the TxD pin; there is an additional 1-bit delay in the transmitter data path due to the data encoding logic.)

Tx Special Bit Pattern Transmission
The DUSCC provides features to transmit special bit patterns (see Table 14).

The TxD pin is held marking after a hardware reset, a reset Tx command, when the transmitter is not enabled, and during underrun/idle, if this feature is selected through TPR[7:5]. The TxD pin is also held marking if the transmitter is enabled, and the TxFIFO is empty (ASYNC), or if a TSOM or TSOMP command has not been issued (SYNC modes).

The following command bits can be appended to characters in the TxFIFO: TEOM, TDLE, exclude from CRC, and reset TxCRC. An invoked command(s) is appended to the next character loaded into the TxFIFO and follows the character through the FIFO until that character is ready to be loaded into the TxSR. The transmitter data path is shown in Figure 10. The following describes the operation of the transmitter for the various protocols.

Tx ASYNC Mode
Serialization begins when the TxFIFO data is loaded into the TxSR. The transmitter first sends a start bit, then the programmed number of bits/character (TPR[1,0]), a parity bit (if specified), and the programmed number of stop bits. Following the transmission of the stop bits, if a new character is not available in the TxFIFO, the TxD output goes to marking and the underrun condition (TRSR[7]) is set.

Table 13. Abort Sequence — Protocol Mode

TPRA [7:6]	PROTOCOL	LAST CHARACTER
00	BOP	FLAG following either FCS (if selected) or last data character
	COP	Last byte of FCS before line begins SYN or MARKing
10	BOP	Abort sequence (11111111) prior to MARKing
	COP	Last byte of FCS before line begins SYN or MARKing
11	BOP	Abort sequence (11111111) prior to FLAG
	COP	First SYN of SYN sequence

Table 14. Special Bit Patterns

PROTOCOL	BIT PATTERN
ASYNC-BREAK	An all 0's character including parity bit (if specified) and stop bits. Used for send break command.
COP-SYN	Contained in S1R (single SYN mode) or in S1R/S2R (dual SYN modes). Used for TSOM and TSOMP commands and for non-transparent mode linefill and IDLE.
COP-DLE	Used for TDLE command and for BISYNC transparent mode linefill and to generate BISYNC control sequences.
COP-CRC	16/8 bits from the CRC/LRC accumulator used for TEOM command or for auto-EOM modes.
BOP-FLAG	01111110. Used for TSOM, TSOMP, and TEOM commands, for auto-EOM modes, and as an IDLE line fill.
BOP-ABORT	11111111. Used for send ABORT command or during TxFIFO underrun.
BOP-CRC	16 bits from the CRC accumulator used for TEOM command or for auto-EOM modes.
BOP/COP MARK	All 1's pattern on data line.

Transmission resumes when the CPU loads a new character into the TxFIFO or issues a send break command. The send break command clears the TxFIFO and forces a continuous space (low) on the TxD output after the character in TxSR (if any) is serialized. A send break acknowledge (TRSR[4]) is returned to the CPU to facilitate reassertion of the send break command in order to send an integral number of break characters. The send break condition is cleared when the reset Tx or disable Tx command is issued.

Tx COP Modes
Transmitter commands associated with all COP modes are: transmit SOM (TSOM, transmit start of message), transmit SOM with PAD (TSOMP), transmit EO' (TEOM, transmit end of message), reset TxCRC, exclude from CRC, and transmit DLE.

A TSOM or send TSOMP command must be issued to start COP transmission. TSOM (without PAD) causes the TxCRC/LRC generator to be initialized and one or two SYN characters from S1R/S2R to be loaded into the TxSR and shifted out on the TxD output. A parity bit, if specified, is appended to each SYN character after the MSB. Send SOM acknowledge (TRSR[4]) is asserted when the SYN output begins. The user may reinvoke the command to cause multiple SYNs to be transmitted. If the command is not reinvoked and the TxFIFO is empty, SYN patterns continue to be transmitted until the TxFIFO is loaded. If data is present in the FIFO, the first character is loaded into the TxSR and serialization of the data begins. Note that the TxFIFO may be preloaded with data before the TSOM is issued.

The TSOMP command causes all characters in the TxFIFO (PAD characters) to be loaded into the TxSR and serialized if the Tx is enabled. Unlike the transmit SOM without PAD command, data (non-PAD characters) cannot be preloaded into the TxFIFO. While the PAD is transmitted, parity is disabled and character length is automatically set to 8 bits regardless of the value in TPR[1:0]. When the TxFIFO becomes empty after the PAD, the TxCRC/LRC generator is initialized, the SYN character(s) are transmitted with optional parity appended, and send SOM acknowledge asserted. Operation then proceeds in the same manner as the TSOM command; the user has the option to invoke the TSOM command to cause multiple SYNs to be transmitted.

After the TSOM/TSOMP command is executed, characters in the TxFIFO are loaded into the TxSR and shifted out with a parity bit, if specified, appended after the MSB. If, after the opening SYN(s) and at least one data has

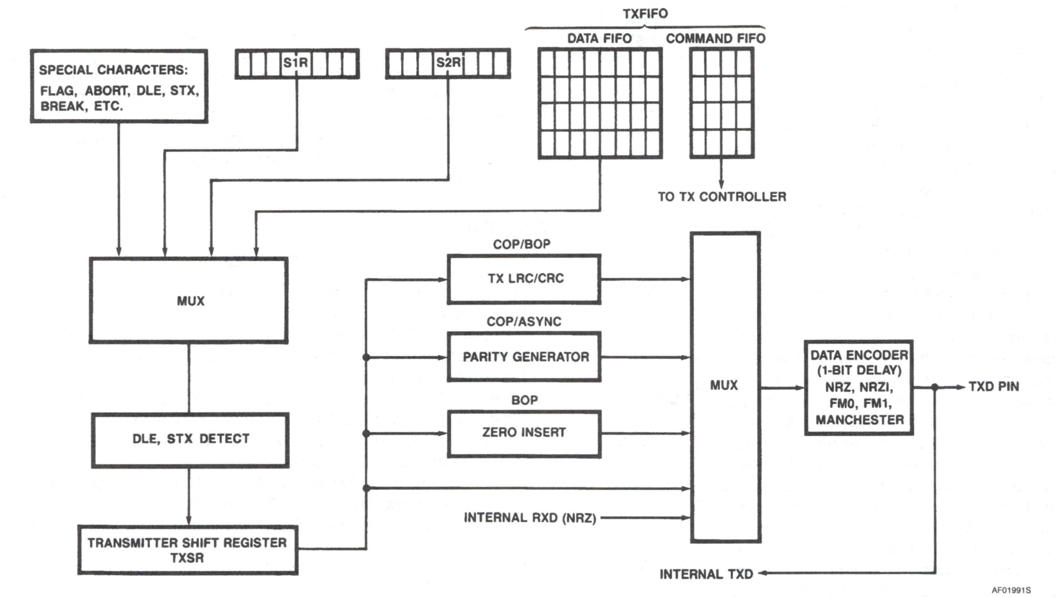


Figure 10. Transmitter Data Path

been transmitted, the TxFIFO is empty, a data underrun condition results and TRSR[7] is asserted. The transmitter's action on data underrun is determined by TPR[7:6] and the COP protocol. If TPR_[7:6] = '10', the transmitter line fills with MARK characters until a character is loaded into the FIFO. If TPR_[7:6] = '11' is selected, the transmitter line fills with SYN, SYN1-SYN2, or DLE-SYN1 for mono-sync, dual sync, and BISYNC transparent modes, respectively. If TPR[7:6] = '00', the BCC characters are transmitted and frame complete (TRSR[6]) is set. TxD then assumes the programmed idle state (TPR[5]) of MARKs or SYN1/SYN1-SYN2.

Operation resumes with the transmission of a SYN sequence when a TSOM command is invoked. A TSOMP command is ignored unless the transmitter is disabled and then reenabled.

An appended TEOM command also terminates the frame as described above. It occurs after transmission of the character to which the TEOM is appended. The TEOM command can be explicitly asserted through the channel command register. If TPR[4] = '1', the TEOM is automatically appended to a character in DMA mode, if the DONEN input is asserted when that character is loaded into the TxFIFO, or if the counter/timer is counting transmitted characters when the character which causes the counter to reach zero count is loaded.

The TDLE command when appended to a character in the TxFIFO, causes the DLE character to be loaded into the TxSR and serialized before the TxFIFO character is loaded into the TxSR and serialized. This feature is particularly useful for BISYNC operation. The DLE character will be excluded from the CRC accumulation in BISYNC transparent mode (see below), but will be included in all other COP modes.

In BISYNC mode, transmission of a DLE-STx character sequence (either via a send TDLE command appended to the STx character, or via DLE and STx loaded into the TxFIFO) puts the transmitter into the transparent text mode of operation. In this mode, normally restricted character sequences can be transmitted as 'normal' bit sequences. The switch occurs after transmission of the two characters, so that the DLE and STx are included in the BCC accumulation. If the DLE-STx is to be excluded from the CRC, the user should issue a 'reset CRC' command prior to loading the next character.

Another method of excluding the two characters from the CRC is to invoke the 'exclude from CRC' command prior to loading the character(s) into the FIFO. While in transparent mode, the transmitter line fills with DLE-SYN1 and automatically transmits an extra DLE if it finds a DLE in the TxFIFO ('DLE stuffing'). The transmitter reverts to non-transparent mode when the frame complete status is set in TRSR[5].

CRC/LRC accumulation can be specified in all COP modes; the type of CRC is specified via CMR2[2:0]. The TSOM/TSOMP commands set the CRC/LRC accumulator to its initial state and accumulation begins with the first non-SYN character after the initial SYN(s) are transmitted. PAD characters are not subject to CRC accumulation. In non-BISYNC or BISYNC normal modes, all transmitted characters except linefill characters (SYNs or MARKs) are subject to accumulation. In BISYNC transparent mode, odd (stuffed) DLEs and the DLE-SYN1 linefill are excluded from the accumulation. Characters can be selectively excluded from the accumulation by invoking the 'exclude from CRC' command prior to loading the character into the FIFO.

Accumulation stops when transmission of the first character of the BCC begins. The CPU can set the accumulator to its initial state prior to the transmission of any character by using the appended reset CRC command. The CRC generator is also automatically initialized after the EOM is sent.

TxBOP Modes

Transmitter commands associated with BOP modes are TSOM, TSOMP, TEOM, and transmit ABORT (TABRK). The TSOM and TSOMP commands are identical to COP modes except that a FLAG character (01111110) is used as the start of message sequence instead of the SYNs, and FLAG(s) that continue to be sent until the TxFIFO is loaded. There is no zero insertion (see below) during transmission of the PAD characters, and they are not preceded by a FLAG or accumulated in the CRC. Character length is automatically set to 8 bits regardless of TPR[1:0].

The first characters loaded into the TxSR from the TxFIFO are the address and control fields, which have fixed character lengths of eight bits. The number of address field bytes is determined by CMR1[4:3]. If extended address field is specified, the field is terminated if the first address octet is H'00' or if the LSB of the octet is a 1. The number of control field bytes is selected by CMR1_[5]. If any information field characters follow the control field (forming an I field), they are transmitted with the number of bits per character programmed in TPR[1:0]. The TEOM command can be appended to the last character either explicitly or automatically as described for COP mode. When the character with the appended TEOM is loaded from the TxFIFO, it is transmitted with the character length specified by OMR[7:5]. In this way, a residual character of 1 – 8 bits is transmitted without requiring the CPU to change the Tx character length for this last character.

After the opening FLAG and first address octet have been transmitted, an underrun occurs (TRSR[7] = 1) if the TxFIFO is empty when the transmitter requires a new character. The underrun control bits (TPR[7:6]) determine whether the transmitter line fills with either ABORT-MARKs, ABORT-FLAGs (see below), or ends transmission with the 'normal' end of message sequence.

EOM on underrun is functionally similar to EOM due to an appended TEOM command. If the EOM is due to underrun, the normal character length applies to the last data character. After the last character is transmitted, the FCS (inverted CRC) and closing FLAG are sent, frame complete (TRSR[5]) is set, and the TxCRC is initialized. If the TxFIFO is empty after the closing FLAG has been sent, TxD will assume the programmed idle state of FLAGs or MARKs (TPR[5]) and wait for a character to be loaded into the FIFO or for a TSOM command to be issued. If the TxFIFO is not empty at that time, the TxFIFO data will be loaded into the TxSR and serialized. In that case, the closing FLAG is the opening FLAG of the next frame.

The user can control the number of FLAGs between frames by invoking the TSOM command after frame complete is asserted. The DUSCC then operates in the same manner as for transmission of multiple FLAGs at the beginning of a frame. When the command is no longer reinvoked, transmission of the TxFIFO data will begin. If the FIFO is empty, FLAGs continue to be transmitted.

The DUSCC provides automatic zero insertion in the data stream to prevent erroneous transmission of the FLAG sequence. All data characters loaded into the TxSR from the TxFIFO and characters transmitted from the CRC generator are subject to zero insertion. For this feature a zero is inserted in the serial data stream each time five consecutive ones (regardless of character boundaries) have been transmitted.

A send ABORT command clears the TxFIFO and inserts an ABORT character of eight ones (not subject to zero insertion) into the TxSR for transmission after the current character has been serialized. A send abort ack (TRSR[4]) facilitates reassertion of send abort by the user to guarantee transmission of multiple abort characters. This feature can be used to send the 15-ones idle sequence. The transmitter sends either marks or FLAGs after the abort character(s) has been transmitted, depending on TPR[7:6]. Operation resumes with the transmission of a FLAG when a TSOM command is invoked. A TSOMP command is ignored unless the transmitter is disabled and then reenabled.

CRC accumulation can be specified in all BOP modes. The type of CRC is specified via CMR2[2:0], and is normally selected as CRC-CCITT preset to ones, although any option is valid. Note that LRC8 option is not allowed in BOP modes.

The TSOM/TSOMP command sets the CRC accumulator to its initial state and accumulation begins with the first address octet after the initial FLAG(s). Accumulation stops when transmission of the first character of the FCS begins. The CPU can set the accumulator to its initial state prior to the transmission of any character by using the appended reset CRC command and can exclude any character from the accumulation by use of the exclude from CRC command, but these features would not normally be used in BOP modes. The CRC generator is also automatically initialized after the EOM or an ABORT are sent.

TxBOP Loop Mode

The loop modes are used by secondary stations on the loop, while the primary station operates in the BOP primary mode. Both the transmitter and receiver must be enabled. Loop operation is initiated by issuing the 'go on-loop' command. The receiver looks for the receipt of seven contiguous ones and then asserts the LC_N output to cause external loop control hardware to put the DUSCC into the loop, with the TxD output echoing the RxD input with a 2-bit time delay. The echoing process continues until a go active on poll (GAP) command is invoked. The DUSCC then looks for receipt of an EOP bit pattern (a zero followed by seven ones, 11111110) and changes the last one of the EOP into a zero making it an opening FLAG. Loop sending (TRSR[6]) is asserted at that same time. The action of the transmitter after sending the initial FLAG depends on the status of the transmit FIFO.

If the transmit FIFO is not empty, a normal frame transmission begins. The operation is then similar to normal BOP operation with the following differences:

1. An ABORT command, an underrun, or receipt of the turnaround sequence (H'00') or a FLAG cause the transmitter to cease operation and to revert to echoing the RxD input with a 2-bit time delay. A new transmission cannot begin until the GAP command is reinvoked and a new EOP sequence is received.
2. Subsequent to sending the EOM sequence of FCS-FLAG, the DUSCC examines the internal GAP flip-flop. If it is not set (having been reset by the 'reset GAP' command), the DUSCC reverts to echoing the received data. If the internal GAP flip-flop is still set, transmission of a new frame begins, with the user having control of sending multiple FLAGs between frames by use of the 'send SOM' command. If the FIFO is empty at this time, the DUSCC continues to send FLAGs until the data is loaded into the FIFO or until GAP is reset. If the latter occurs, it reverts to echoing RxD.

   When the DUSCC reverts to echoing RxD in any of the above cases, the last transmitted zero and seven ones will form an EOP for the next station down the loop.

   If the TxFIFO is empty when the EOP is recognized, the transmitter continues to send FLAGs until there is data in the FIFO. If a turnaround sequence or the reset GAP command is received before the FIFO is loaded, the transmitter switches to echoing RxD without any data transmission. Otherwise a frame transmission begins as above when a character is loaded into the FIFO. The mechanism provides time for the CPU to examine the received frame (the frame preceding the EOP) to determine if it should respond or not, while holding its option to initiate a transmission.

   Termination of operation in the loop mode should be accomplished by use of the 'go off-loop' command. When the command is invoked, the DUSCC looks for the receipt of eight contiguous ones. It then negates the LC_N output to cause the external loop control hardware to remove the DUSCC from the loop without affecting operation of other units remaining on the loop.

RECEIVER

The receiver data path includes two 9-bit holding registers, HSRH and HSRL, an 8-bit character comparison register, CCSR, two synchronizing flip flops, a receiver shift register, RxSR, the programmable SYN comparison registers, S1R and S2R, and BISYNC character comparison logic. The DUSCC configures this circuitry and utilizes it according to the operational mode selected for the channel through the two mode registers CMR1 and CMR2. For all data paths, character data is assembled according to the character bit count, in the RxSR, and is moved to the RxFIFO with any appended statuses when assembly is completed. Figure 1 depicts the four data paths created in the DUSCC for the previous protocols.

Receiver RxFIFO, RxRDY

The receiver converts received serial data on RxD (LSB first) into parallel data according to the transmission format programmed. Data is shifted through a synchronizing flip flop and one or more shift registers, the last of which is the 8-bit receiver shift register (RxSR). Bits are shifted into the RxSR on the rising edge of each 1× receive clock until the LSB is in RxSR[0]. Hence, the received character is right justified, with all unused bits in the RxSR cleared to zero. A receive character length counter generates a character boundary signal for synchronization of character assembly, character comparisons, break detection (ASYNC), and RxSR to RxFIFO transfers (except for BOP residual characters). During COP and BOP hunt phases, the SYN/FLAG comparison is made each receive bit time, as abort, and idle comparisons in BOP modes.

An internal clock from the BRG, the DPLL or the counter/timer, or an external 1× or 16× clock may be used as the receiver clock in ASYNC mode. The BRG or counter/timer cannot be used directly for the receiver clock in synchronous modes, since these modes require a 1× receive clock that is in phase with the received data. This clock may come externally from the RTxC or TRxC pins, or it may be derived internally from the DPLL. Encoded data is internally converted to NRZ format for the receiver circuits by using clock pulses generated by the DPLL.

When a complete character has been assembled in the RxSR, it is loaded into the receive FIFO with appended status bits. The most significant data bits of the character are set to zero if the character length is less than eight bits. In ASYNC and COP modes the user may select, via RPR[3], whether the data transferred to the FIFO includes the received parity bit or not. The receiver indicates to the CPU or DMA controller that it has data in the FIFO by asserting the channel's RxRDY status bit (GSR[4] or GSR[0]) and, if in DMA mode, the corresponding receiver DMA request pin.

The RxFIFO consists of four 8-bit holding registers with appended status bits for character count complete indications (all modes), character compare indication (ASYNC), EOM indication (BISYNC/BOP), and parity, framing, and CRC errors. Data is loaded into the RxFIFO from the RxSR and extracted (read) by the CPU or DMA controller via the data bus. An RxFIFO read creates an empty RxFIFO position for new data from the RxSR.

RxRDY assertion depends on the state of OMR[3]:

1. If OMR[3] is 0 (FIFO not empty), RxRDY is asserted each time a character is transferred from the receive shift register to the receive FIFO. If it is not reset by the CPU, RxRDY remains asserted until the receive FIFO becomes empty, at which time it is automatically negated. If it is reset by the CPU, it will remain negated, regardless of the current state of the receive FIFO, until a new character is transferred from the RxSR to the RxFIFO.
2. If OMR[3] is 1 (FIFO full), RxRDY is asserted:
   a. when a character transfer from the receive shift register to the receive FIFO causes it to become full
   b. when a character with a tagged EOM status bit is loaded into the FIFO (BISYNC or BOP) regardless of RxFIFO full condition.
   c. when the counter/timer is programmed to count received characters and the character which causes it to reach zero count is loaded into the FIFO (ICTSR [6]).
   d. when the beginning of a break is detected in ASYNC mode regardless of the RxFIFO full condition.

If it is not reset by the CPU, RxRDY remains asserted until the FIFO becomes empty, at which time it is automatically negated. If it is reset by the CPU, it will remain negated regardless of the current state of the receive FIFO, until it is asserted again due to one of the above conditions.

The assertion of RxRDY causes an interrupt to be generated if IER[4] and the channel's master interrupt enable (ICR[0] or ICR[1]) are asserted.

When DMA operation is programmed, the RxRDY status bit is routed to the DMA control circuitry for use as the channel receiver DMA request. Assertion of RxRDY results in assertion of RTxDRQN output.

Several status bits are appended to each character in the RxFIFO. When the FIFO is read, causing it to be 'popped', the status bits associated with the new character at the top of the RxFIFO are logically ORed into the RSR. Therefore, the user should read R SR before reading the RxFIFO in response to RxRDY activation. If character-by-character status is desired, the RSR should be read and cleared each time a new character is received. The user may elect to accumulate status over several characters or over a frame by clearing RSR at appropriate times. This mode would normally also be used when operating in DMA mode. If the RxFIFO is empty when a read is attempted, and wait mode as specified in CMR2[5:3], is not being used, a H'FF' is output on the data bus.

In all modes, the DUSCC protects the contents of the FIFO and the RxSR from overrun. If a character is received while the FIFO is full and a character is already in the RxSR waiting to be transferred into the FIFO, the overrunning character is discarded and the OVERRUN status bit (RSR[5]) is asserted. If the overrunning character is an end-of-message character, the character is lost but the FIFOed EOM status bit will be asserted when the character in the RxSR is loaded into the FIFO.

Operation of the receiver is controlled by the enable receiver command. When this command is issued, the DUSCC goes into the search for start bit state (ASYNC), search for SYN state (COP modes), or search for FLAG state (BOP modes). When the disable receiver command is issued, the receiver ceases operation immediately. The RxFIFO is cleared on master reset, or by a reset receiver command. However, disabling the receiver does not affect the RxFIFO, RxRDY, or DMA request operation.

Receiver DCD and RTS Controls

If DCD enable Rx, RPR[2], is asserted, the DCD input must be asserted for the receiver to operate. If RPR[2] is asserted and the sampling circuit detects that the DCD input has been negated, the receiver ceases operation immediately. Operation resumes when the sampled DCD is asserted again. A change of state detector is provided on the DCD input of each channel. The required duration of the DCD level change is described in the discussion of ICTSR[5]. The user may program a change of state to cause an interrupt to be generated (master interrupt enable ICR[0] or [1] and IER [7] must be set) so that appropriate action can be taken.

In ASYNC mode, RPR[4] can be programmed to control the deactivation of the RTS_N output by the receiver. RTS_N can be manually asserted and negated by writing to OMR_[0]. However, the assertion of RPR[4] causes RTS to be negated automatically upon receipt of a valid start bit if the channel's receive FIFO is already full. When this occurs, the RTSN negated status bit, RSR[6], is set. This may be used as a flow control feature to prevent overrun in the receiver by using the RTS_N output signal to control the CTS_N input of the remote transmitter. The new character will be assembled in the RxSR, but its transfer to the FIFO will be delayed until the CPU reads the FIFO, making the FIFO position available for the new character.

Once enabled, receiver operation depends on channel protocol mode. The following describes the receiver operation for the various protocols:

RxASYNC Mode

When first enabled, the receiver goes into the search for start bit state, looking for a high-to-low (mark-to-space) transition of the start bit on the RxD input. If a transition is detected, the state of the RxD pin is sampled again each 16× clock for 7½ clocks (16× clock mode) or at the next rising edge of the bit time clock (1× clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again.

If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals (16 periods of the 16× Rx clock; one period of the 1× Rx clock) at the theoretical center of the bit, until the proper number of data bits and the parity bit (if specified) have been assembled, and the first stop bit has been detected.

The assembled character is then transferred to the RxFIFO with appended parity error (if parity is specified) and framing error status bits. The DUSCC can be programmed to compare this character to the contents of S1R. The appended character compare status bit, RSR[7], is set if the data matches and there is no parity error.

After the stop bit is sampled, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (i.e. framing error) and RxD remains low for one-half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

If a break condition is detected (RxD low for entire character time including optional parity and first stop bit), only one character consisting of all zeros will be loaded into the RxFIFO and break start detect, RSR[2], will be set. The RxD input must return to a high condition for at least one half of a bit time (16× clock mode) or for one bit time (1× clock mode) before the break condition is terminated and the search for the next start bit begins. At that time, the break end detect condition, RSR[3], is set.

Rx COP Modes

When the receiver is enabled in COP modes, it first goes into the SYN hunt phase, testing the received data each bit time for receipt of the appropriate SYN bit pattern, plus parity if specified, to establish character boundaries. Receipt of the SYN bit pattern terminates hunt phase and places the receiver in the data phase, in which all leading SYNs are stripped and the RxFIFO begins to load starting with the first non-SYN character. In COP single SYN protocol mode, S1R contains the SYN character required to establish character synchronization. In COP dual SYN and BISYNC protocol modes, S1R and S2R contain the first and second SYN characters, respectively, required to establish character synchronization. The SYN character length is the same as the character length programmed in RPR[1:0], plus the parity bit if parity is specified. SYN characters received with a parity error, when parity is specified, are considered invalid and will not cause synchronization to be achieved.

If external synchronization is programmed (RPR[4] = 1), the internal SYN detection and special character recognition logic are disabled and receipt of SYN characters is not required. A pulse on the SYNI input pin will establish character synchronization and terminate hunt phase. The SYNI pin is ignored after the first input on the SYNIN pin is received. The receiver must be disabled and then reenabled to resynchronize or to return to normal mode. This must be programmed in conjunction with CMR1[2:0] = 110. Refer to the description of RPR[4] for further details.

The SYN detect status bit, RSR[2], is set whenever SYN1, SYN1-SYN2, or DLE-SYN1 is detected for single SYN, dual SYN/BISYNC normal, and BISYNC transparent modes, respectively, and the SYNOUT pin will go active for one receive clock period one bit time after SYN detection in HSRH/HSRL. After character sync has been attained, the receiver enters the data phase and assembles characters in the RxSR, beginning with the first non-SYN character, with the least significant bit received first. It computes the BCC if specified, checks parity if specified, and checks for overrun errors.

The operation of the BCC (CRC/LRC) logic depends on the particular COP mode in use. The BCC is initialized upon first entering the data phase. For non-BISYNC modes, all received characters after entering data phase are included in the BCC computation, except for leading SYNs and SYNs which are specified to be stripped by RPR[7]. As each received character is transferred from the RxSR to the FIFO, the current value of the BCC characters is checked and the CRC ERROR status bit (RSR[1]) is set if the value of the CRC remainder is not the expected value. The EOM status bit, RSR[7], is not set since there is no defined end-of-message character. The receiver computes the BCC for text messages automatically when operating in BISYNC protocol mode.

BISYNC Features

The DUSCC provides support for both BISYNC normal and transparent operations. The following summarizes the features provided. Both EBCDIC and ASCII text messages can be handled by the DUSCC as selected by CMR1[5]. The receiver has the capability of recognizing special characters for the BISYNC protocol mode (see Table 15).

All sequences in Table 15, except SOH and STx, when detected explicitly cause a status to be affected. The following describes the conditions when this occurs.

The first character received when entering data phase for a header or text message should be an SOH, an STx, or a DLE-STx two-character sequence. Receipt of any of these initializes the CRC generator and starts the CRC accumulation. The SOH places the receiver in header mode, receipt of the STx places it in text mode, and receipt of the DLE-STx sequence (at any time) automatically places the receiver in transparent mode and sets the XPNT mode status bit, TRSR[0]. There is no explicit status associated with SOH and STx. If any other characters are received when entering the data phase, the message is treated as a control message and will not be accumulated in CRC.

After the data phase is established, the receiver searches the data stream for an end of message control character(s):

Header field: ENQ, ETB, or ITB

Normal text field: ENQ, ETx, ETB, or ITB

Transparent text field: DLE-ENQ, DLE-ETx, DLE-ETB, or DLE-ITB

Control message field: EOT, NAK, ACK0, ACK1, WACK, RVI or TTD

Detection of any one of these sequences causes the EOM status bit, RSR[7], to be set. Also if RPR[5] is set and the receiver does not detect a closing PAD (four 1's) after the 'EOT' or 'NAK', the PAD error status bit, RSR[6], is set. When the abort sequence ENQ or DLE-ENQ is detected, the character is tagged with an EOM status and transferred to the FIFO, but the appended CRC error status bit should be ignored. For the other EOM control sequences, the receiver waits for the next two bytes (the CRC bytes) to be received, checks the value of the CRC generator, and tags the transferred character with a CRC error, RSR[1], if the CRC remainder is not correct. See Figure 11 for an example of BCC accumulation in various BISYNC messages.

The CRC bytes are normally not transferred to the FIFO, unless the transfer FCS to FIFO control bit, RPR[6], is asserted. In this case the EOM and CRC error status bits will be tagged onto the last byte of the last FCS byte instead of to the last character of the message. After detecting one of the end-of-message (EOM) character sequences and setting RSR[7], the receiver automatically goes into auto hunt mode for the SYNC characters and PAD check if RRP[5] is set.

SYN Pattern Stripping

Leading SYNs (before a message) are always stripped and excluded from the FCS, but SYN patterns within a message are treated by the receiver according to the RPR[7] bit. SYN character patterns are defined for the various COP modes as follows:

COP single SYN mode — SYN1

COP dual SYN mode — SYN1, and SYN2 when immediately preceded by SYN1.

BISYNC normal mode — SYN1, and SYN2 when immediately preceded by SYN1. SYN1 is always stripped, even if it is not followed by SYN2 when stripping is selected.

BISYNC transparent mode — DLE - SYN1, where the DLE is the last of an odd number of consecutive DLEs.

0	Strip only RPR[7] leading the SYN and do not accumulate in FCS.
1	Strip all SYNs. Additionally, strip odd DLEs when operating in BISYNC transparent mode. Do not accumulate stripped characters in FCS.

Processing of the SYN patterns is determined by the RPR[7] bit, the COP mode, and the position of the pattern in the frame. This is summarized in Table 16.

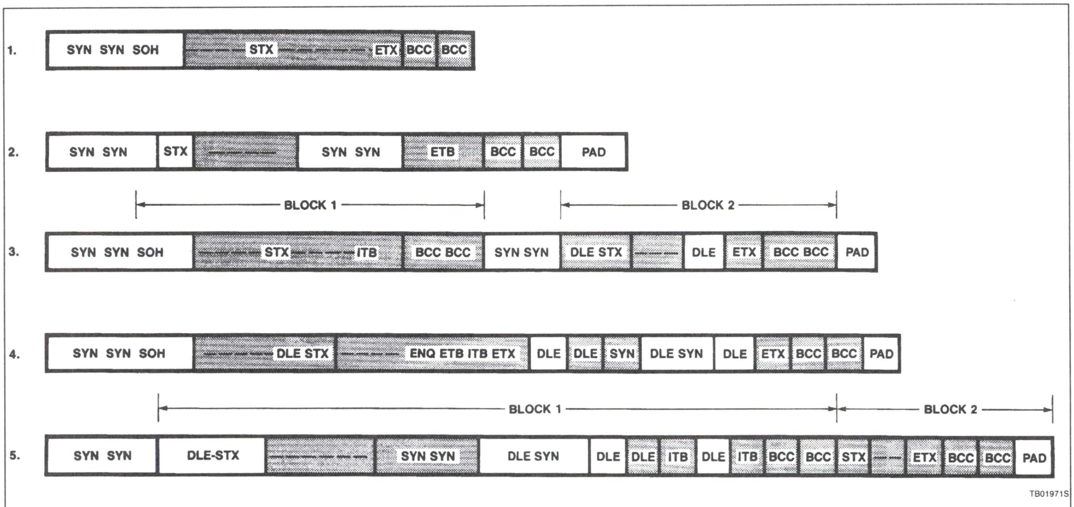
The value of the RPR[7] field does not affect the setting of the SYN DETECT status bit, RSR[2], and the generation of a SYNOUT pulse when a SYN pattern is received.

RxBOP Mode

In BOP protocol mode, the receiver may be in any one of four phases: hunt phase, address field (A) phase, control field (C) phase, or information field (I) phase. The character length for the A and C phases is always 8 bits. The I field character length is specified in RPR[1:0].

DESIGN NOTE: If the residual character length is not zero, the unused most significant bits in the receiver FIFO are not necessarily zero. The unused bits should be ignored, this will not cause a CRC error.

After an enable receiver command is executed, the receiver enters hunt phase, in which a comparison for the string (01111110) is done every Rx bit time. The FLAG delineates the beginning (and end) of a received frame and establishes the character boundary. Each FLAG match in CCSR causes the FLAG detect status bit (RSR[2]) to be set and SYNOUT N pin to be activated one bit time later for one receive clock period. FLAGs with an overlapping zero will be detected. All FLAGs are deleted from the data stream.



NOTES:
1. The BCC accumulator is stopped after the end of message character sequence is accumulated. All shaded areas are accumulated.
2. ENQ (DLE - ENQ) in a text message should be treated as an abort.
3. Opening SYNs will be stripped by the receiver.

Figure 11. Example of BCC Accumulation in Various BISYNC Messages

Table 15. BISYNC Features

BISYNC — Single-Character Sequences

Sequence	ASCII	EBCDIC	Description
SOH	H'01	H'01	Start of header
STx	H'02	H'02	Start of text
ETx	H'83	H'03	End of text
EOT	H'04	H'37	End of transmission
ENQ	H'85	H'2D	Enquiry
DLE	H'10	H'10	Data link escape
NAK	H'15	H'3D	Negative ack
ETB	H'97	H'26	End of transmission block
ITB	H'1F	H'1F	End of intermediate transmission block

BISYNC — Two-Character Sequences

Sequence	ASCII	EBCDIC	Description
ACK0	H'10,B0	H'10,70	Acknowledge 0
ACK1	H'10,31	H'10,61	Acknowledge 1
WACK	H'10,38	H'10,6B	Wait before transmit positive ack
RVI	H'10,BC	H'10,7C	Reverse interrupt
TTD	H'02,85	H'02,2D	Temporary text delay

BISYNC — (Transparent Text Mode) — Two-Character Sequences

Sequence	ASCII	EBCDIC	Description
DLE-ENQ	H'10,85	H'10,2D	Enquiry
DLE-ITB	H'10,1F	H'10,1F	End of intermediate transmission block
DLE-ETB	H'10,97	H'10,26	End of transmission block
DLE-ETx	H'10,83	H'10,03	End of text
DLE-STx	H'10,02	H'10,02	Start of transparent text mode

Table 16. SYN Pattern Processing

MODE	RPR [7]	LEADING SYNs	WITHIN A MESSAGE
BISYNC	0	no FCS no FIFO	no FCS Pattern into FIFO
	1	no FCS no FIFO	no FCS no FIFO
COP	0	no FCS	Accumulate in FCS
		no FIFO	Pattern into FIFO
	1	no FCS no FIFO	no FCS no FIFO

Once a FLAG has been detected, the receiver will exit hunt phase and enter address phase. The handling of the address field is determined by the values programmed in CMR1[2:0], which selects one of the BOP modes. The BOP secondary address modes are selected by CMR1 [4:3] and function as in the description that follows.

Single-Octet Address
For receive, the address comparison for a secondary station is made on the first octet following the opening FLAG. A match occurs if the first octet after the FLAG matches the contents of S1R, or if all parties address (RPR[3]) is asserted and the first octet is equal to H'FF'.

Dual-Octet Address
For receive, the address comparison for a secondary station is made on the first two octets following the opening FLAG.

A match occurs if the first two octets after the FLAG match the contents of S1R and S2R respectively, or if all parties address (RPR[3]) is asserted and the first two octets are equal to H'FF, FF'.

Dual Address with Group Mode
For receive, the address comparison for a secondary station is made on the first two octets following the opening FLAG. A match occurs for one of three possible conditions. If the first two octets after the FLAG match the contents of S1R and S2R, respectively, or if the first octet is H'FF' and the second matches the contents of S2R (group mode), or when all parties address (RPR[3]) is asserted and the first two octets are equal to H'FF, FF'. The second condition (group mode) allows a selected group of stations to receive a message.

Extended Address Mode
Extend address field to the next octet if the LSB of the current address octet is zero. Address field is terminated if the LSB of the address is a one. The address field will be terminated after the first octet if the null address H'00' is received/transmitted as the first address octet. For this mode the receiver does not perform an address comparison (all received characters after the opening FLAG are transferred to the FIFO) but does determine when the address field is terminated.

The length of the A field may be a single octet, a dual octet, or more octets, as described above. A primary station or an extended address secondary station does not perform an address comparison, and all characters in the A, C, and I fields after the flag are transferred to the FIFO. Although address field comparisons are not performed, the length of the address field is still determined by CMR1[4:3]. For the other secondary address modes, if there is a match, or the received character(s) match either of the other enabling conditions (group or all-parties address), all characters in the A, C, and I fields are transferred to the FIFO. If there is no match, the receiver returns to the FLAG hunt phase.

C phase begins after A phase is terminated. The receiver receives one or two control characters, CMR1[5]. After this phase is terminated, the character length is switched automatically from 8 bits to the number of bits specified in RPR[1:0] and the information field phase is entered.

The frame is terminated when a closing FLAG is detected. The same FLAG can also serve as the opening FLAG of the next frame. The 16 bits received prior to the closing FLAG form the frame check sequence (if an FCS is specified in CMR2[2:0]). All non-FLAG characters of the frame are accumulated in the CRC checker and the result is compared to the expected remainder. Failure to match will cause a CRC error. EOM detect RSR[7], RCL not zero RSR[0], and CRC error RSR[1] are normally FIFOed with the last character of the I field. RCL not zero RSR[0] is set if the length of the last character of the I field does not have the length programmed in RPR[1:0]. The residual character length in TRSR[2:0] is also valid at that time. The CRC characters themselves are normally not passed to the RxFIFO. However, if the transfer FCS to FIFO control bit RPR[6] is asserted, the FCS bytes will be transferred to the FIFO. In this case the EOM, CRC error, and RCL not zero status bits will be tagged onto the last byte of the CRC sequence instead of to the last character of the message.

If the closing FLAG is received prior to receipt of the appropriate number of A field, C field as programmed in CMR1[5:3], and FCS field octets, a short frame will be detected and RSR[4] will be set. The I field need not be present in a valid frame. An abort (11111110) comparison is done after an opening FLAG has been received and up to receipt of the closing FLAG. A match causes the abort detect status bit (RSR[6]) to be set. The receiver then enters FLAG search mode. The abort is stripped from the received data stream.

If a zero followed by 15 contiguous ones is detected, the idle detect status bit RSR[3] is set. This comparison is done whenever the receiver is enabled. Therefore, it can occur before or after a received frame.

Zero deletion is performed during BOP receive. A zero after 5 contiguous ones is deleted from the data stream regardless of character boundaries. Deleted zeroes are not subject to CRC accumulation. FLAG, ABORT, and IDLE comparisons are done prior to zero deletion.

If external synchronization is programmed (RPR[4] = 1), the internal FLAG detection and address comparison logic is disabled and receipt of FLAGs is not required. In this arrangement, a pulse on the SYNI-N input pin will establish synchronization and terminate hunt phase. The receiver will then go immediately into the I-field mode with zero deletion disabled, assembling and transferring characters into the FIFO with the character length specified in RPR[1:0]. The SYNI-N pin is ignored after the first input on the SYNI-N pin is received. The receiver must be disabled and then reenabled to resynchronize or to return to normal operating mode.

This mode must be programmed in conjunction with CMR1[2:0] = 110. Refer to the description of RPR[4] for further details.

BOP Loop Mode

Operation of the receiver in BOP loop protocol mode is similar to operation in other BOP modes, except that only certain frame formats are supported. Several character detection functions that interact with the operation of the transmitter or transmitter commands are added:

1. When the 'go on-loop' command is invoked, the receiver looks for the receipt of a zero followed by seven ones and then asserts the LC_N output.
2. When the 'go off-loop' command is invoked, the receiver looks for the receipt of eight contiguous ones and then negates the LC_N output.
3. The TxD output normally echoes the receive input with a two bit time delay. When the 'go active on poll' command is asserted, the receiver looks for an EOP (a zero followed by seven ones) and then switches the TxD output line to the normal transmitter output. Receipt of an EOP or an ABORT sets RSR[6].
4. Receipt of a turnaround sequence (eight contiguous zeros) terminates the transmitter operation, if any, and returns the TxD output to echoing the RxD input. RSR[3] is set if a turnaround is received.

See transmitter operation for additional details.

SUMMARY OF COP FEATURES

COP Dual SYN Mode	
SYN detect	SYN1-SYN2
Linefill	SYN1-SYN2
SYN stripping	SYN1-SYN2 used to establish character sync, i.e. leading SYNs. Subsequent to this (after receiving first non-SYN character), SYN1 and SYN1-SYN2 if stripping is specified by RPR[7].
Excluded from FCS**	SYN1 and SYN1-SYN2 before beginning of message, i.e. leading SYNs and, if SYN stripping is specified by RPR[7] anywhere else in the message for the Rx; linefill SYN1-SYN2 for Tx regardless of RPR[7]. (If SYN stripping is not specified, then SYNs within a message will be included in FCS by Rx.)
BISYNC normal mode	
SYN detect	SYN1-SYN2
Linefill	SYN1-SYN2
SYN stripping	SYN1-SYN2 used to establish character sync, i.e. leading SYNs. Subsequent to this (after receiving first non-SYN character), SYN1 and SYN1-SYN2 if stripping is specified by RPR[7].
Excluded from FCS	All SYNs either before or within a message, regardless of RPR[7], plus additional characters as required by the protocol.
BISYNC transparent mode	
SYN detect	*DLE-SYN1
Linefill	DLE-SYN1
SYN/DLE stripping	*DLE-SYN1 and odd DLEs if stripping is specified by RPR[7].
Excluded from FCS	*DLE-SYN1 and odd DLEs, regardless of RPR[7] plus additional characters as required by the protocol.
COP single SYN mode	
SYN detect	SYN1
Linefill	SYN1
SYN stripping	SYN1 used to establish character sync, i.e. leading SYNs. Subsequent to this, SYN1 if stripping is specified by RPR[7].
Excluded from FCS**	SYN1 before beginning of message, i.e. leading SYNs, and if SYN stripping is specified by RPR[7], anywhere else in the message for the Rx; linefill SYN1 for Tx regardless of RPR[7]. (If SYN stripping is not specified, then SYNs within a message will be included in FCS by Rx.)

NOTES:

*DLE indicates last DLE of an odd number of consecutive DLEs.

**In non-BISYNC COP modes (single or dual SYN case), if SYN stripping is off, i.e. RPR[7] = 0, then SYNs within a message will be included in FCS by receiver. Therefore, the remote DUSCC transmitter should be careful not to let the TxFIFO underrun since the linefill SYN characters are not accumulated in FCS by the transmitter regardless of RPR[7]. Letting the TxFIFO underrun will result in a CRC error in the receiver.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	−55 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to +70°C, $V_{CC} = 5.0V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage					
	All except X1/CLK				0.8	V
	X1/CLK				.04	V
V_{IH}	Input high voltage					
	All except X1/CLK		2			V
	X1/CLK		2.4			V
V_{OL}	Output low voltage					
	except DONEN, IRQN	$I_{OL} = 2.4mA$			0.5	V
	DONEN, IRQN	$I_{OL} = 8.8mA$			0.5	V
V_{OH}	Output high voltage					
	(except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{XIL}	X1/CLK low input current	$V_{IN} = 0$, X2 grounded	−4.5			mA
		$V_{IN} = 0$, X2 floated	−4.5			mA
I_{XIH}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 grounded			1	mA
		$V_{IN} = V_{CC}$, X2 floated			30	mA
I_{X2L}	X2 low input current	$V_{IN} = 0$, X1/CLK floated	−100			μA
I_{X2H}	X2 high input current	$V_{IN} = V_{CC}$, X1/CLK floated			100	μA
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-State leakage current	$V_0 = 0$ to V_{CC}	−10		10	μA
I_{OC}	Open-collector output leakage current	$V_0 = 0$ to V_{CC}	−10		10	μA
I_{CC}	Power supply current				200	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other conditions above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 35°C/W junction to ambient for ceramic DIP, 35°C/W for plastic DIP, and 41°C/W for PLCC.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all inputs except X1/CLK swing between 0.4V and 2.4V with a transition time of 20ns maximum. For X1/CLK, this swing is between 0.4V and 4.4V. All time measurements are referenced at input voltages of 0.8V and 2V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. Test condition for outputs: $C_L = 150pF$, except interrupt outputs. Test conditions for interrupt outputs: $C_L = 50pF$, $R_L = 2.7k\Omega$ to V_{CC}.
8. This specification will impose maximum 68000 CPU CLK to 6MHz. Higher CPU CLK can be used if repeating bus reads are not performed.
9. Execution of the valid command (after it is latched) requires 3-4 falling edges of X1 (see Figure 14).
10. Tests for open drain outputs are intended to guarantee switching of the output transistor. Measurement of this response is referenced from the midpoint of the switching signal to a point 0.5V above V_{OL}. This point represents noise margin that assures true switching has occurred. Beyond this level, the effects of external circuitry and test environment are pronounced and can greatly affect the resultant measurement.

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5V ±5%[4, 5, 6, 7]

NO.	FIGURE	PARAMETER	TENTATIVE LIMITS			UNIT
			Min	Typ	Max	
1	12	RESETN pulse width	3			µs
2	13, 15	A0 – A6 setup time to CSN low	10			ns
3	13, 15	A0 – A6 hold time from CSN high	0			ns
4	13, 15	RWN setup time to CSN low	0			ns
5	13, 15	RWN hold time to CSN high	0			ns
6	13, 15	CSN high pulse width[8]	160			ns
7	13, 15, 16	CSN or IACKN high from DTACKN low	30			ns
8	13, 16	Data valid from CSN low			300	ns
9	13	Data bus floating from CSN high			270	ns
10	15	Data hold time from CSN high[9, 10, 11]	0			ns
11	13, 16	DTACKN low from read data ready	0			ns
12	13, 15	DTACKN low from CSN low			480	ns
13	13, 15	DTACKN high from CSN high			235	ns
14	13, 15	DTACKN high impedance from CSN high			360	ns
15	16	DTACKN low from IACKN low			650	ns
16	17	Port input setup time to CSN low	20			ns
17	17	Port input hold time from CSN low	120			ns
18	17	Port output valid from DTACK low			560	ns
		IRQN high from:[12]				
19	18	Read RxFIFO (RxRDY interrupt)			300	ns
		Write TxFIFO (TxRDY interrupt)			300	ns
		Write RSR (receiver condition interrupt)			650	ns
		Write TRSR (receiver/transmitter interrupt)			650	ns
		Write ICTSR (port change and timer/counter interrupt)			650	ns
20	19	X1/CLK high or low time	25			ns
		X1/CLK frequency	2	14.7456	16	MHz
		CTCLK high or low time	100			ns
		CTCLK frequency	0		4	MHz
		RxC high or low time	110			ns
		RxC frequency (16×)	0		4	MHz
		(1×)	0		4	MHz
		TxC high or low time	110			ns
		TxC frequency (16×)	0		4	MHz
		(1×)	0		4	MHz
21	20	TxD output from 1X TxC input low			385	ns
22	20	TxD output from TxC output low	0		50	ns
23	21	RxD data setup time to RxC high	50			ns
24	21	RxD data hold time from RxC high	0			ns
25	22	IACKN low to daisy chain low			350	ns
26	24	Data valid from receive DMA ACKN			320	ns
27	23, 24	DTCN width	100			ns
28	23, 24	RDYN low to DTCN low	80			ns
29	24	Data bus float from DTCN low			300	ns
30	23, 24	DMA ACKN low to RDYN (DTACKN) low			600	ns
31	23, 24	RDYN high from DTCN low			420	ns
32	23, 24	RDYN high impedance from DTCN low			530	ns
33	24	Receive DMA REQN high from DMA ACKN low			585	ns
34	24	Receive DMA ACKN width	150			ns
35	23, 24	Receive DMA ACKN low to DONEN low			300	ns
36	23	Data setup to DTCN low	300			ns
37	23	Data hold from DTCN low	230			ns
38	23	Transmit DMA REQN high from ACKN low			550	ns
39	23	Transmit DMA ACKN deasserted width	150			ns
40	23	Transmit DMA ACKN low to DONEN low output			550	ns
41	25	CSN low to transmit DONEN low output			400	ns
42	25	CSN low to transmit DMA REQ negated			620	ns
43	25	CSN low to receive DONEN low			400	ns
44	25	CSN low to receive DMA REQ negated			620	ns

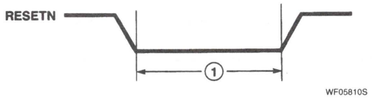

RESETN
①
WF05810S

Figure 12. Reset Timing

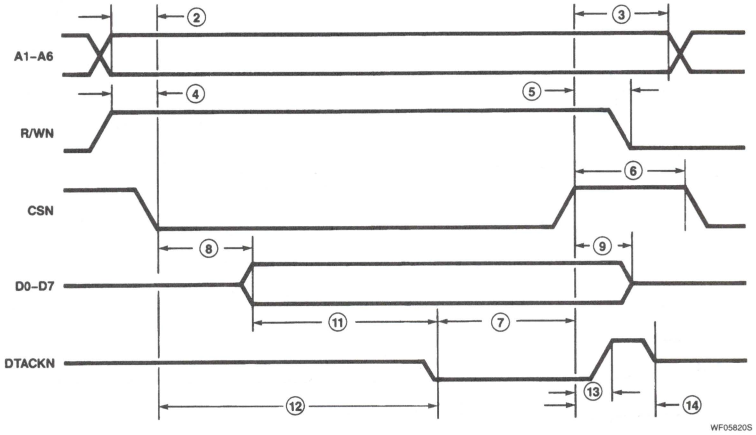

A1–A6
R/WN
CSN
D0–D7
DTACKN
②
③
④
⑤
⑥
⑦
⑧
⑨
⑪
⑫
⑬
⑭
WF05820S

Figure 13. Bus Timing (Read Cycle)

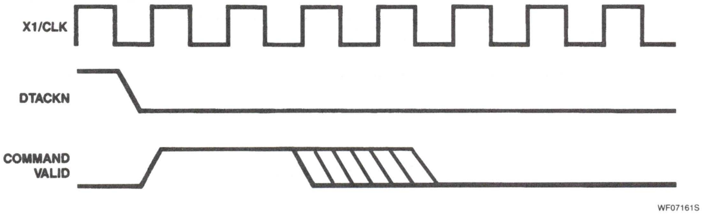

X1/CLK
DTACKN
COMMAND VALID
WF07161S

Figure 14. Command Timing

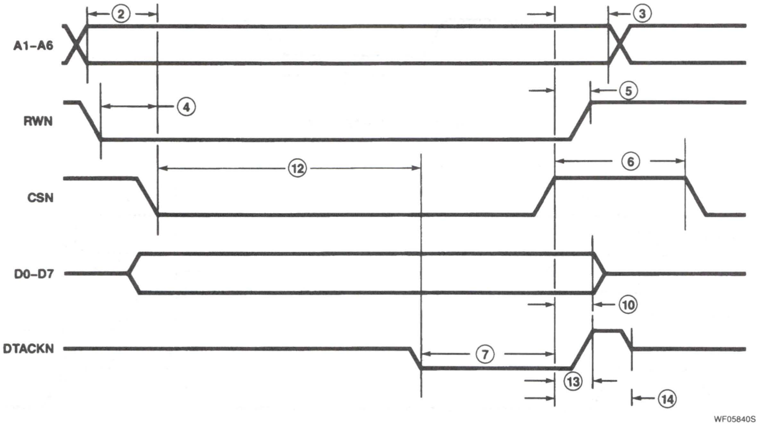

A1–A6
RWN
CSN
D0–D7
DTACKN
②
③
④
⑤
⑥
⑦
⑩
⑫
⑬
⑭
WF05840S

Figure 15. Bus Timing (Write Cycle)

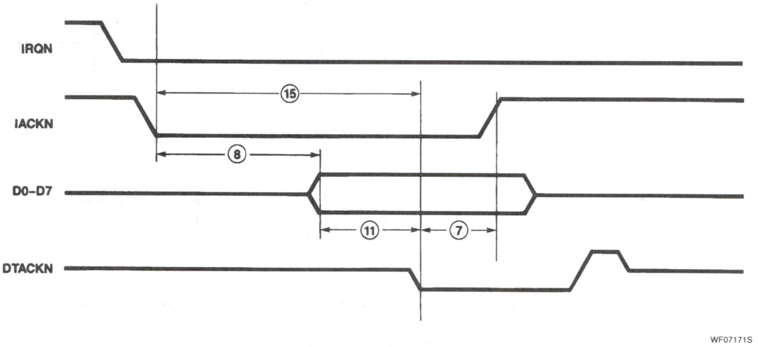

IRQN
IACKN
D0–D7
DTACKN
⑮
⑧
⑪
⑦
WF07171S

Figure 16. Interrupt Cycle Timing

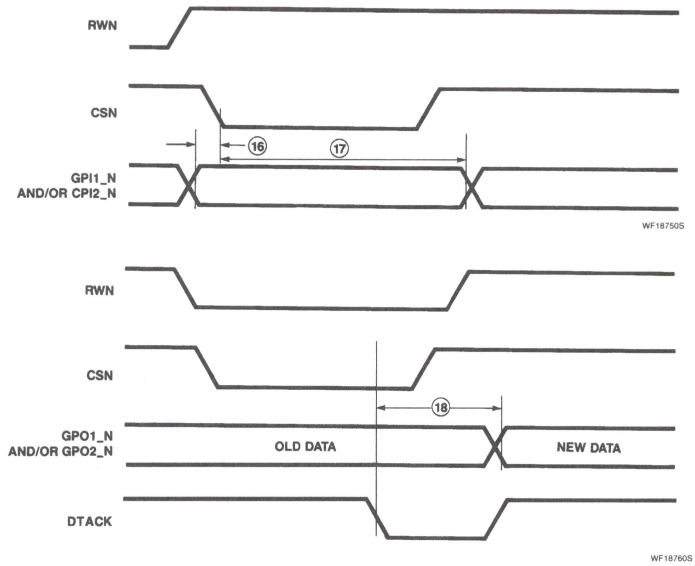

RWN
CSN
⑯
⑰
GPI1_N
AND/OR CPI2_N
WF18750S
RWN
CSN
⑱
GPO1_N
AND/OR GPO2_N
OLD DATA
NEW DATA
DTACK
WF18760S

Figure 17. Port Timing

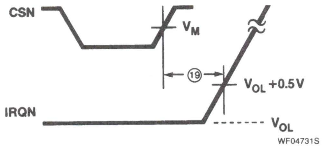

CSN
V_M
⑲
V_{OL} +0.5V
IRQN
V_{OL}
WF04731S

Figure 18. Interrupt Timing

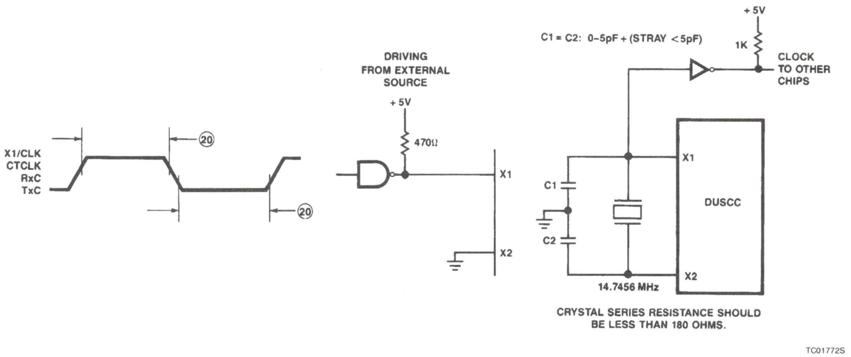

X1/CLK
CTCLK
RxC
TxC
⑳
⑳
DRIVING
FROM EXTERNAL
SOURCE
+5V
470Ω
X1
X2
C1 = C2: 0-5pF + (STRAY <5pF)
+5V
1K
CLOCK
TO OTHER
CHIPS
X1
C1
DUSCC
C2
X2
14.7456 MHz
CRYSTAL SERIES RESISTANCE SHOULD
BE LESS THAN 180 OHMS.
TC01772S

Figure 19. Clock Timing

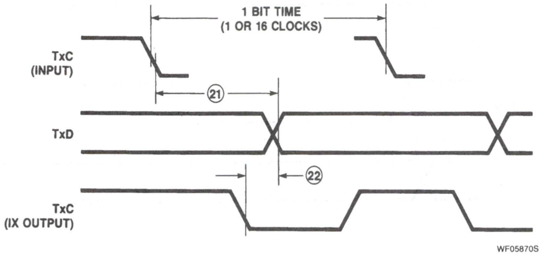

1 BIT TIME
(1 OR 16 CLOCKS)
TxC
(INPUT)
㉑
TxD
㉒
TxC
(IX OUTPUT)
WF05870S

Figure 20. Transmit Timing

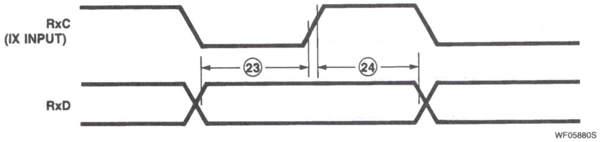

RxC
(IX INPUT)
㉓
㉔
RxD
WF05880S

Figure 21. Receive Timing

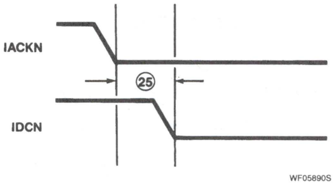

IACKN
㉕
IDCN
WF05890S

Figure 22. Interrupt Daisy Chain Timing

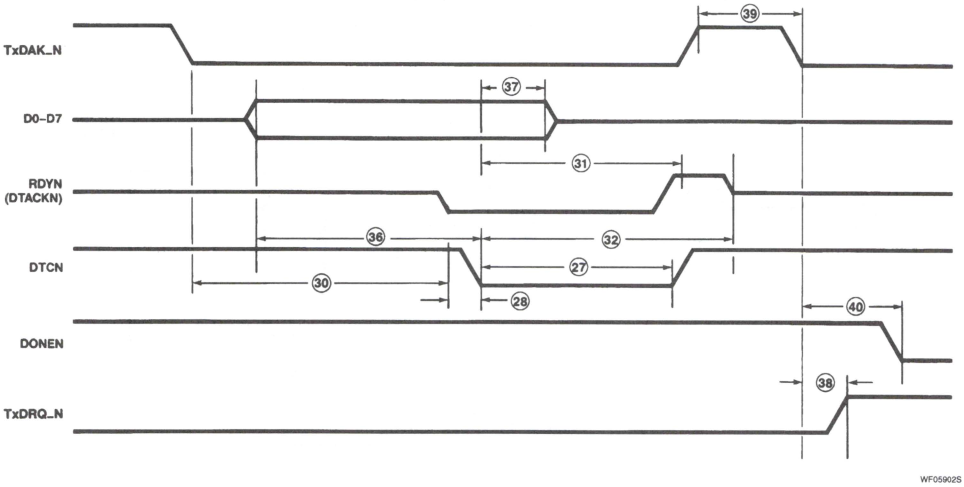

TxDAK_N
㊴
D0–D7
㊲
㉛
RDYN
(DTACKN)
㊱
㉜
DTCN
㉗
㉚
㉘
㊵
DONEN
㊳
TxDRQ_N
WF05902S

Figure 23. DMA Transmit Write Timing — Single Address DMA Mode

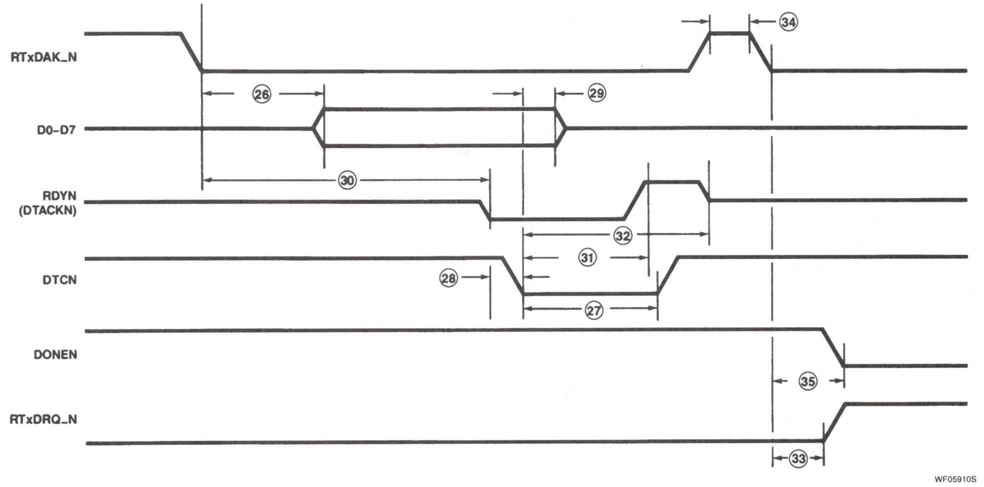

RTxDAK_N
D0–D7
RDYN
(DTACKN)
DTCN
DONEN
RTxDRQ_N
㉞
㉖
㉙
㉚
㉜
㉛
㉘
㉗
㉟
㉝
WF05910S

Figure 24. DMA Receiver Read Timing — Single Address DMA Mode

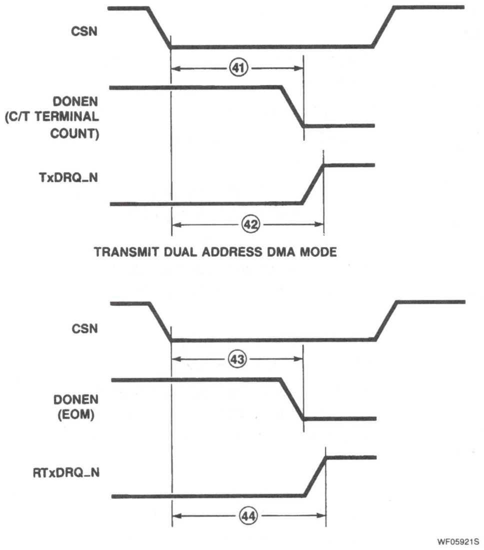

CSN
㊶
DONEN
(C/T TERMINAL
COUNT)
TxDRQ_N
㊷
TRANSMIT DUAL ADDRESS DMA MODE
CSN
㊸
DONEN
(EOM)
RTxDRQ_N
㊹
WF05921S

Figure 25. Dual Address DMA Mode Timing

Signetics

Microprocessor Products

SCN68681
Dual Asynchronous Receiver/ Transmitter (DUART)

Product Specification

DESCRIPTION

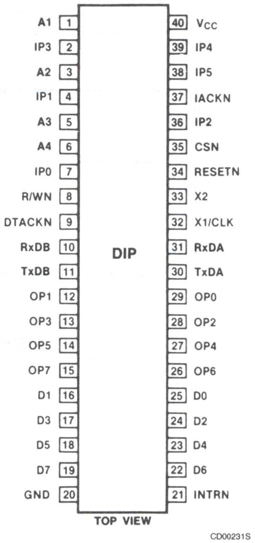
The Signetics SCN68681 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single-chip MOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It is compatible with other S68000 family devices, and can also interface easily with other microprocessors. The DUART can be used in polled or interrupt driven systems.

The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16× clock derived from a programmable counter/timer, or an external 1× or 16× clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

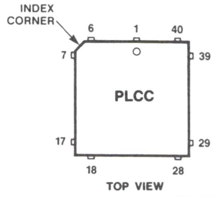
FEATURES

- **S68000 bus compatible**
- **Dual full-duplex asynchronous receiver/transmitter**
- **Quadruple buffered receiver data registers**
- **Programmable data format**
  - **5 to 8 data bits plus parity**
  - **Odd, even, no parity or force parity**
  - **1, 1.5 or 2 stop bits programmable in 1/16-bit increments**
- **Programmable baud rate for each receiver and transmitter selectable from:**
  - **18 fixed rates: 50 to 38.4k baud**
  - **One user-defined rate derived from programmable timer/ counter**
  - **External 1× or 16× clock**
- **Parity, framing, and overrun error detection**
- **False start bit detection**
- **Line break detection and generation**
- **Programmable channel mode**
  - **Normal (full-duplex)**
  - **Automatic echo**
  - **Local loopback**
  - **Remote loopback**
- **Multi-function programmable 16-bit counter/timer**
- **Multi-function 6-bit input port**
  - **Can serve as clock or control inputs**
  - **Change-of-state detection on four inputs**
- **Multi-function 8-bit output port**
  - **Individual bit set/reset capability**
  - **Outputs can be programmed to be status/interrupt signals**
- **Versatile interrupt system**
  - **Single interrupt output with eight maskable interrupting conditions**
  - **Interrupt vector output on interrupt acknowledge**
  - **Output port can be configured to provide a total of up to six separate wire-ORable interrupt outputs**
- **Maximum data transfer: 1× - 1MB/sec, 16× - 125kB/sec**
- **Automatic wake-up mode for multidrop applications**
- **Start-end break interrupt/status**
- **Detects break which originates in the middle of a character**
- **On-chip crystal oscillator**
- **TTL compatible**
- **Single +5V power supply**

PIN CONFIGURATIONS



CD00231S



CD0044PS

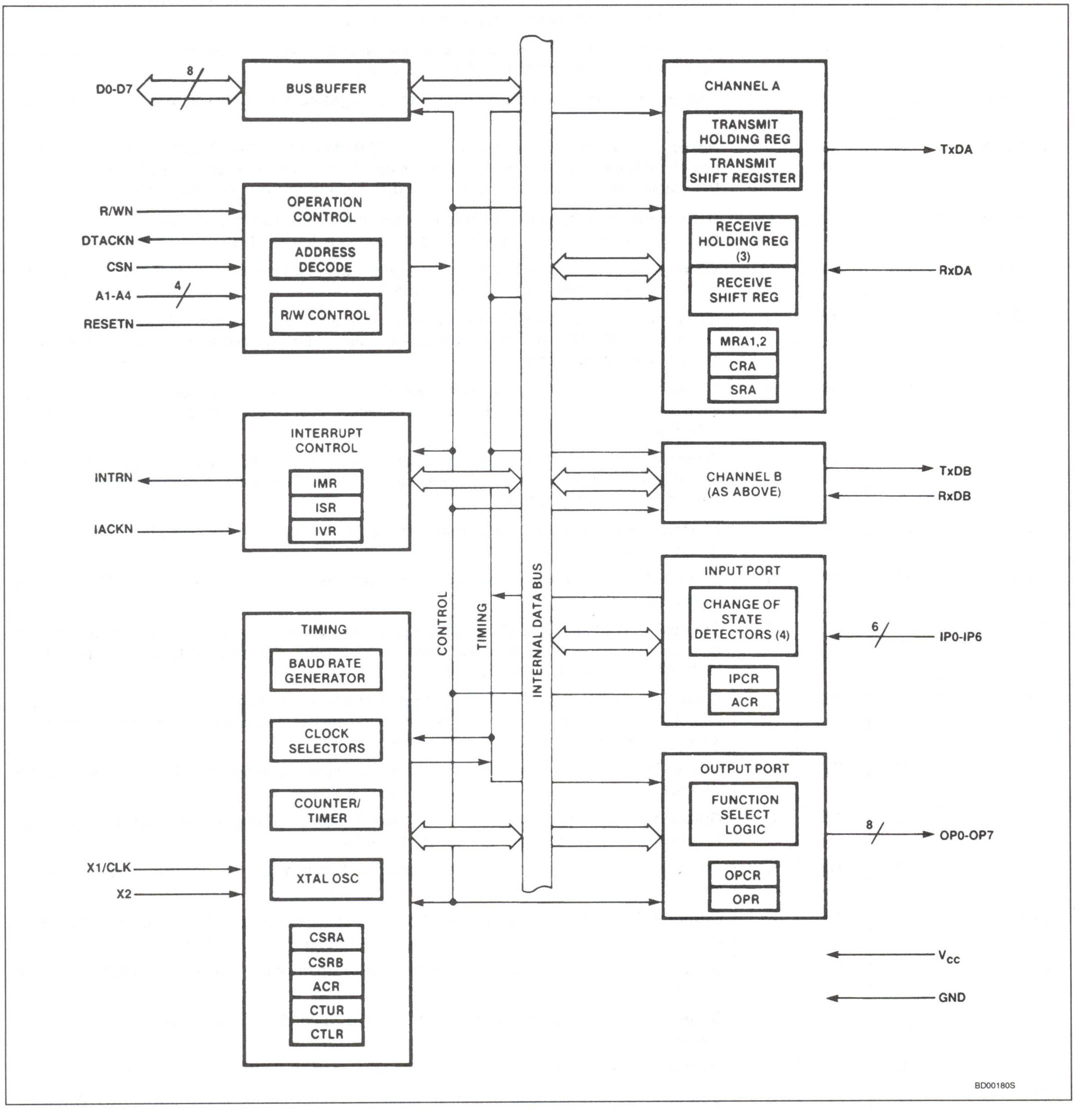
Pin	Function	Pin	Function
1	NC	23	NC
2	A1	24	INTRN
3	IP3	25	D6
4	A2	26	D4
5	IP1	27	D2
6	A3	28	D0
7	A4	29	OP6
8	IP0	30	OP4
9	R/WN	31	OP2
10	DTACKN	32	OP0
11	RxDB	33	TxDA
12	NC	34	NC
13	TxDB	35	RxDA
14	OP1	36	X1/CLK
15	OP3	37	X2
16	OP5	38	RESET
17	OP7	39	CSN
18	D1	40	IP2
19	D3	41	IACKN
20	D5	42	IP5
21	D7	43	IP4
22	GND	44	V_{CC}

Dual Asynchronous Receiver/Transmitter (DUART) SCN68681

ORDERING INFORMATION

PACKAGES	V_{CC} = +5V ±5%, T_A = 0°C to +70°C
Ceramic DIP	SCN68681C1I40
Plastic DIP	SCN68681C1N40
Plastic LCC	SCN68681CIA44

BLOCK DIAGRAM


D0-D7
8
BUS BUFFER
R/WN
DTACKN
CSN
A1-A4
4
RESETN
OPERATION CONTROL
ADDRESS DECODE
R/W CONTROL
INTRN
IACKN
INTERRUPT CONTROL
IMR
ISR
IVR
TIMING
BAUD RATE GENERATOR
CLOCK SELECTORS
COUNTER/ TIMER
X1/CLK
X2
XTAL OSC
CSRA
CSRB
ACR
CTUR
CTLR
CONTROL
TIMING
INTERNAL DATA BUS
CHANNEL A
TRANSMIT HOLDING REG
TRANSMIT SHIFT REGISTER
RECEIVE HOLDING REG (3)
RECEIVE SHIFT REG
MRA1,2
CRA
SRA
TxDA
RxDA
CHANNEL B (AS ABOVE)
TxDB
RxDB
INPUT PORT
CHANGE OF STATE DETECTORS (4)
IPCR
ACR
6
IP0-IP6
OUTPUT PORT
FUNCTION SELECT LOGIC
OPCR
OPR
8
OP0-OP7
Vcc
GND
BD00180S

Dual Asynchronous Receiver/Transmitter (DUART) SCN68681

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
D0 – D7	25, 16, 24 17, 23, 18 22, 19	28, 18, 27 19, 26, 20 25, 21	I/O	**Data Bus:** Bidirectional 3-State data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CSN	35	39	I	**Chip Select:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0 – D7 as controlled by the R/WN and A1 – A4 inputs. When high, places the D0 – D7 lines in the 3-State condition.
R/WN	8	9	I	**Read/Write:** A high input indicates a read cycle and a low input indicates a write cycle, when a cycle is initiated by assertion of the CSN input.
A1 – A4	1, 2, 5, 6	2, 4, 6, 7	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESETN	34	38	I	**Reset:** A low clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), initializes the IVR to hex 0F, puts OP0 – OP7 in the high state, stops the counter/timer, and puts channel A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
DTACKN	9	10	O	**Data Transfer Acknowledge:** 3-State active low output asserted in write, read, or interrupt cycles to indicate proper transfer of data between the CPU and the DUART.
INTRN	21	24	O	**Interrupt Request:** Active low, open drain output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
IACKN	37	41	I	**Interrupt Acknowledge:** Active low input indicating an interrupt acknowledge cycle. In response, the DUART will place the interrupt vector on the data bus and will assert DTACKN if it has an interrupt pending.
X1/CLK	32	36	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. If a crystal is used, a capacitor must be connected from this pin to ground (see Figure 7).
X2	33	37	I	**Crystal 2:** Connection for other side of the crystal. If a crystal is used, a capacitor must be connected from this pin to ground (see Figure 7). If X1/CLK is driven from an external source, this pin should be grounded.
RxDA	31	35	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
RxDB	10	11	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	30	33	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	11	13	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	29	32	O	**Output 0:** General purpose output, or channel A request to send (RTSAN, active low); can be deactivated automatically on receive or transmit.
OP1	12	14	O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low); can be deactivated automatically on receive or transmit.
OP2	28	40	O	**Output 2:** General purpose output, or channel A transmitter 1× or 16× clock output, or channel A receiver 1× clock output.
OP3	13	15	O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1× clock output, or channel B receiver 1× clock output.
OP4	27	30	O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.
OP5	14	16	O	**Output 5:** General purpose output or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	26	29	O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output.
OP7	15	17	O	**Output 7:** General purpose output, or channel B open drains, active low, TxRDYB output.
IP0	7	8	I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN).

Dual Asynchronous Receiver/Transmitter (DUART) SCN68681

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
IP1	4	5	I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN).
IP2	36	40	I	**Input 2:** General purpose input, or channel B receiver external clock input (RxCB), or counter/timer external clock input. When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP3	2	3	I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP4	39	43	I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	38	42	I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
V_{CC}	40	44	I	**Power Supply:** +5V supply input.
GND	20	22	I	**Ground**

Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

Also provided on the SCN68681 are a multipurpose 6-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

BLOCK DIAGRAM

The SCN68681 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The DTACKN output is asserted during write and read cycles to indicate to the CPU that data has been latched on a write cycle, or that valid data is present on the bus on a read cycle.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurrence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR), the interrupt status register (SR), the auxiliary control register (ACR), and the interrupt vector register (IVR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. When IACKN is asserted, and the DUART has an interrupt pending, the DUART responds by placing the contents of the IVR register on the data bus and asserting DTACKN.

Outputs OP3 – OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864nHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART. If an external signal is used instead of a crystal, X1 should be driven using a configuration similar to the one in Figure 7.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4k baud. The clock outputs from the BRG are at 16× the actual baud rate. The counter/timer can be used as a timer to produce a 16× clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection, for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Communications Channels A and B

Each communications channel of the SCN68681 comprises a full-duplex asynchronous receiver/transmitter (DUART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The input port pulse detection circuitry uses a 38.4kHz sampling clock derived from one of the baud rate generator taps. This results in a sampling period of slightly more than 25μs (assuming that the clock input is 3.6864MHz). The detection circuitry, in order to guarantee a true change in level has occurred, requires that two successive samples at the new logic level be observed. As a consequence, the minimum duration of the signal change is 25μs if the transition occurs coincident with the first sample pulse. The 50μs time refers to the situation in which the change of state is just missed and the first change of state is not detected until 25μs later.

The inputs to this unlatched 6-bit port can be read by the CPU by performing a read operation at address D16. A high input results in a logic 1 while a low input results in a logic 0. D7 will always be read as a logic 1 and D6 will reflect the level of IACKN. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A High-to-Low or Low-to-High transition of these inputs lasting longer than 25–50μs will set the corresponding bit in the input port change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice-versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E_{16} with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F16 with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The SCN68681 is conditioned to transmit data when the transmitter is enabled through the command register. The SCN68681 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted, if it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted; only if the transmitter has been disabled.

Receiver

The SCN68681 is conditioned to receive data when enabled through the command register. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16× clock for 7½ clocks (16× clock model) or at the next rising edge of the bit time clock (1× clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical-OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the

Table 1. 68681 Register Addressing

A4	A3	A2	A1	READ (R/WN = 1)	WRITE (R/WN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	Rx Holding Register A (RHRA)	Tx Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	Rx Holding Register B (RHRB)	Tx Holding Register B (THRB)
1	1	0	0	Interrupt Vector Reg. (IVR)	Interrupt Vector Reg. (IVR)
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit (SP[4]) will be set upon receipt of the start bit of the new (overrunning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to '11' for channels A and B respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, with receivers that are normally disabled, examine the received data stream and 'wakeup' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program in the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in Table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x switches the pointer to MR2x. the pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to Table 2 for register bit descriptions. The reserved registers at addresses H'02' and H'0A' should never be read during normal operation since they are reserved for internal diagnostics.

Table 2. Register Bit Formats

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RxRTS CONTROL	**RxINT SELECT**	**ERROR MODE**	**PARITY MODE**		**PARITY TYPE**	**BITS PER CHARACTER**	
MR1A **MR1B**	0=no 1=yes	0=RxRDY 1=FFULL	0=char 1=block	00 = with parity 01 = force parity 10 = no parity 11 = multi-drop mode		0=even 1=odd	00=5 01=6 10=7 11=8	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	CHANNEL MODE		**TxRTS CONTROL**	**CTS ENABLE Tx**	**STOP BIT LENGTH***			
MR2A **MR2B**	00=Normal 01=Auto echo 10=Local loop 11=Remote loop		0=no 1=yes	0=no 1=yes	0=0.563 1=0.625 2=0.688 3=0.750	4=0.813 5=0.875 6=0.938 7=1.000	8=1.563 9=1.625 A=1.688 B=1.750	C=1.813 D=1.875 E=1.938 F=2.000

NOTE:
*Add 0.5 to values shown for 0–7 if channel is programmed for 5 bits/char.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVER CLOCK SELECT				**TRANSMITTER CLOCK SELECT**			
CSRA **CSRB**	See Text				See Text			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
		MISCELLANEOUS COMMANDS			**DISABLE Tx**	**ENABLE Tx**	**DISABLE Rx**	**ENABLE Rx**
CRA **CRB**	not used– must be 0	See Text			0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVED BREAK	**FRAMING ERROR**	**PARITY ERROR**	**OVERRUN ERROR**	**TxEMT**	**TxRDY**	**FFULL**	**RxRDY**
SRA **SRB**	0=no 1=yes *	0=no 1=yes *	0=no 1=yes *	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes

NOTE:
*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits 4:0. These bits are cleared by a 'reset error status' command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	OP7	**OP6**	**OP5**	**OP4**	**OP3**		**OP2**	
OPCR	0 = OPR[7] 1=TxRDYB	0 = OPR[6] 1=TxRDYA	0 = OPR[5] 1=RxRDY/ FFULLB	0 = OPR[4] 1=RxRDY/ FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB(1 ×) 11 = RxCB(1 ×)		00 = OPR[2] 01 = TxCA(16 ×) 10 = TxCA(1 ×) 11 = RxCA(1 ×)	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	BRG SET SELECT	**COUNTER/TIMER MODE AND SOURCE**			**DELTA IP3 INT**	**DELTA IP2 INT**	**DELTA IP1 INT**	**DELTA IP0 INT**
ACR	0 = set1 1 = set2	See Table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

Table 2. Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IPCR	**ΔIP3**	**ΔIP2**	**ΔIP1**	**ΔIP0**	**IP3**	**IP2**	**IP1**	**IP0**
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ISR	**INPUT PORT CHANGE**	**Δ BREAK B**	**RxRDY/ FFULLB**	**TxRDYB**	**COUNTER READY**	**Δ BREAK A**	**RxRDY/ FFULLA**	**TxRDYA**
	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IMR	**IN. PORT CHANGE INT**	**Δ BREAK B INT**	**RxRDY/ FFULLB INT**	**TxRDYB INT**	**COUNTER READY INT**	**Δ BREAK A INT**	**RxRDY/ FFULLA INT**	**TxRDYA INT**
	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTUR	**C/T[15]**	**C/T[14]**	**C/T[13]**	**C/T[12]**	**C/T[11]**	**C/T[10]**	**C/T[9]**	**C/T[8]**

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTLR	**C/T[7]**	**C/T[6]**	**C/T[5]**	**C/T[4]**	**C/T[3]**	**C/T[2]**	**C/T[1]**	**C/T[0]**

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IVR	**IVR[7]**	**IVR[6]**	**IVR[5]**	**IVR[4]**	**IVR[3]**	**IVR[2]**	**IVR[1]**	**IVR[0]**

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control

This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select

This bit selects either the channel A receiver ready status (RxRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the accumulation (logical-OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select

If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select

This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits per Character Select
This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select
Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission; i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control
This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Verify the message is sent by waiting until the transmit ready status (TxRDY) is asserted. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-Send Control
If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select
This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of $\frac{9}{16}$ to 1 and $1\frac{9}{16}$ to 2 bits, in increments of $\frac{1}{16}$ bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, $1\frac{1}{16}$ to 2 stop bits can be programmed in increments of $\frac{1}{16}$ bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1× clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select
This field selects the baud rate clock for the channel A receiver as follows:

CSRA[7:4]	Baud Rate Clock = 3.6864MHz ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4k	19.2k
1 1 0 1	Timer	Timer
1 1 1 0	IP4 — 16×	IP4 — 16×
1 1 1 1	IP4 — 1×	IP4 — 1×

The receiver clock is always a 16 × clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select
This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

CSRA[3:0]	Baud Rate ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP3 — 16×	IP3 — 16×
1 1 1 1	IP3 — 1×	IP3 — 1×

The transmitter clock is always a 16× clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select
This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

CSRB[7:4]	Baud Rate ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP2 — 16×	IP2 — 16×
1 1 1 1	IP2 — 1×	IP2 — 1×

The receiver clock is always a 16× clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select
This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

CSRB[3:0]	Baud Rate ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP5 — 16×	IP5 — 16×
1 1 1 1	IP5 — 1×	IP5 — 1×

The transmitter clock is always a 16× clock except for CSRB[3:0] = 1111

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting; e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands
The encoded value of this field may be used to specify a single command as follows:

CRA[6:4]	COMMAND
0 0 0	No command.
0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO pointers are reset.
0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
1 1 0	Start break. Forces the TxDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
1 1 1	Stop Break. The TxDA line will go high (marking) within two bit times. TxDA will remain high for one bit time before the next character, if any, is transmitted.

CRA[3] — Disable Channel A Transmitter
This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter
Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver
This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver
Enables operation of the channel A receiver. If not in the special wake-up mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting; e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break
This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received: further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1× clock).

When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error
This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error
This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error
This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA)
This bit will be set when the channel A transmitter underruns; i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA)
This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

SRA[1] — Channel A FIFO Full (FFULLA)
This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full; i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full. FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA)
This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, only if (after this read) there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select
This bit programs the OP7 output to provide one of following:
- The complement of OPR[7]
- The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select
This bit programs the OP6 output to provide one of the following:
- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select
This bit programs the OP5 output to provide one of the following:
- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select
This bit programs the OP4 output to provide one of the following:
- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select
This field programs the OP3 output to provide one of the following:
- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open-drain output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1× clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1× clock is output.
- The 1× clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1× clock is output.

OPCR[1:0] — OP2 Output Select
This field programs the OP2 output to provide one of the following:
- The complement of OPR[2]
- The 16× clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be 1× clock if CSRA[3:0] = 1111.
- The 1× clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1× clock is output.
- The 1× clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1× clock is output.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select
This bit selects one of two sets of baud rates to be generated by the BRG:

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05k, 1.2k, 2.4k, 4.8k, 7.2k, 9.6k, and 38.4k baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k, and 19.2k baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in Table 3.

ACR[6:4] — Counter/Timer Mode and Clock Source Select
This field selects the operating mode of the counter/timer and its clock source as shown in Table 4.

ACR[3:0] — IP3, IP2, IP1, IP0 Change of State Interrupt Enable
This field selects which bits of the Input Port Change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

Table 3. Baud Rate Generator Characteristics Crystal or Clock = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16X CLOCK (KHz)	ERROR (PERCENT)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2k	307.2	0
38.4k	614.4	0

NOTE:
Duty cycle of 16× clock is 50% ±1%

Table 4. ACR [6:4] Field Definition

ACR [6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)[1]
0 0 1	Counter	TxCA — 1× clock of channel A transmitter
0 1 0	Counter	TxCB — 1× clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)[1]
1 0 1	Timer	External (IP2) divided by 16[1]
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

NOTE:
1. In these modes, the channel B receiver clock should normally be generated from the baud rate generator.

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change of State

These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State

These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initalized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status

This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

ISR[6] — Channel B Change in Break

This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B 'reset break change interrupt' command.

ISR[5] — Channel B Receiver Ready or FIFO Full

The function of this bit is programmed by MR1B[6]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full; i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[4] — Channel B Transmitter Ready

This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready

In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change in Break

This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready or FIFO Full

The function of this bit is programmed by MR1A[6]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full; i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready
This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3–OP7 or the reading of the ISR.

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs respectively of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 0002_{16}. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A4–A1 = 1110) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR. The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A4–A1 = 1111). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

On power-up and after reset, the timer/counter runs in timer mode and can only be restarted. Because it cannot be shut off or stopped, and runs continuously in timer mode, it is recommended that at initialization, the output port, OP3, should be masked off through the OPCR[3:2] = 00 until the T/C is programmed to the desired operational state.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8 bits to the upper 8 bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

IVR — Interrupt Vector Register

This register contains the interrupt vector. The register is initialized to H'0F' by RESET. The contents of the register are placed on the data bus when the DUART responds to a valid interrupt acknowledge cycle.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +6.0	V

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2			V
V_{IH}	Input high voltage (X1/CLK)		4			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	µA
I_{LL}	Data bus 3-State leakage current	$V_O = 0.4$ to V_{CC}	−10		10	µA
I_{X1L}	X1/CLK low input current	$V_{IN} = 0$, X2 grounded	−4	−2	0	mA
		$V_{IN} = 0$, X2 floated	−3	−1.5	0	mA
I_{X1H}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 grounded	−1	0.2	1	mA
		$V_{IN} = V_{CC}$, X2 floated	0	3.5	10	mA
I_{X2L}	X2 low input current	$V_{IN} = 0$, X1/CLK floated	−100	−30	0	µA
I_{X2H}	X2 high input current	$V_{IN} = V_{CC}$, X1/CLK floated	0	+30	100	µA
I_{OC}	Open-collector output leakage current	$V_O = 0.4$ to V_{CC}	−10		10	µA
I_{CC}	Power supply current				150	mA

NOTES:

4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all inputs except X1/CLK swing between 0.4V and 2.4V with a transition time of 20ns maximum. For X1/CLK, this swing is between 0.4V and 4.4V. All time measurements are referenced at input voltages of 0.8V and 2V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. Test condition for outputs: $C_L = 150pF$, except interrupt outputs. Test condition for interrupt outputs: $C_L = 50pF$, $R_L = 2.7k\Omega$ to V_{CC}.
8. This specification will impose maximum 68000 CPU CLK to 6MHz. Higher CPU CLK can be used if repeating bus reads are not performed. Consecutive write operations to the same command register require at least three edges of the X1 clock between writes.
9. This specification imposes a lower bound on CSN and IACKN low, guaranteeing that it will be low for at least 1 CLK period. This requirement is made on CSN only to insure assertion of DTACKN and not to guarantee operation of the part.
10. This specification is made only to insure that DTACKN is asserted with respect to the rising edge of the X1/CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the setup time is violated, DTACKN may be asserted as shown, or may be asserted one clock cycle later.

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4, 5, 6, 7]

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Typ	Max	
Reset timing (Figure 1)					
t_{RES}	RESETN pulse width	1.0			μs
Bus timing (Figures 2, 3, 4)					
t_{AS}	A1–A4 setup time to CSN low	10			ns
t_{AH}	A1–A4 hold time from CSN high	0			ns
t_{RWS}	RWN setup time to CSN high	0			ns
t_{RWH}	RWN holdup time to CSN high	0			ns
t_{CSW}[8]	CSN high pulse width	160			ns
t_{CSD}[9]	CSN or IACKN high from DTACKN low	20			ns
t_{DD}	Data valid from CSN or IACKN low			175	ns
t_{DF}	Data bus floating from CSN or IACKN high			100	ns
t_{DS}	Data setup time to CLK high	100			ns
t_{DH}	Data hold time from CSN high	0			ns
t_{DAL}	DTACKN low from read data valid	0			ns
t_{DCR}	DTACKN low (read cycle) from CLK high			125	ns
t_{DCW}	DTACKN low (write cycle) from CLK high			125	ns
t_{DAH}	DTACKN high from CSN or IACKN high			100	ns
t_{DAT}	DTACKN high impedance from CSN or IACKN high			125	ns
t_{CSC}[10]	CSN or IACKN setup time to clock high	90			ns
Port timing (Figure 5)					
t_{PS}	Port input setup time to CSN low	0			ns
t_{PH}	Port input hold time from CSN high	0			ns
t_{PD}	Port output valid from CSN high			400	ns
Interrupt reset timing (Figure 6)					
t_{IR}	INTRN, or OP3–OP7 when used as interrupts, negated from:				
	Read RHR (RxRDY/FFULL interrupt)			300	ns
	Write THR (TxRDY interrupt)			300	ns
	Reset command (delta break interrupt)			300	ns
	Stop C/T command (counter interrupt)			300	ns
	Read IPCR (input port change interrupt)			300	ns
	Write IMR (clear of interrupt mask bit)			300	ns
Clock timing (Figure 7)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2	3.6864	4	MHz
t_{CTC}	CTCLK high or low time	100			ns
f_{CTC}	CTCLK frequency	0		4	MHz
t_{RX}	RxC high or low time	220			ns
f_{RX}	RxC frequency (16×)	0		2	MHz
	(1×)	0		1	MHz
t_{TX}	TxC high or low time	220			ns
f_{TX}	TxC frequency (16×)	0		2	MHz
	(1×)	0		1	MHz
Transmitter timing (Figure 8)					
t_{TXD}	TxD output delay from TxC low			350	ns
t_{TCS}	Output delay from TxC low to TxD data output			150	ns
Receiver timing (Figure 9)					
t_{RXS}	RxD data setup time to RxC high	240			ns
t_{RXH}	RxD data hold time from RxC high	200			ns

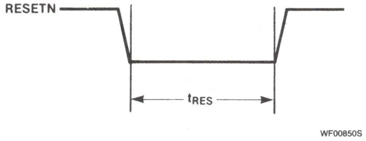

RESETN
t_{RES}
WF00850S

Figure 1. Reset Timing

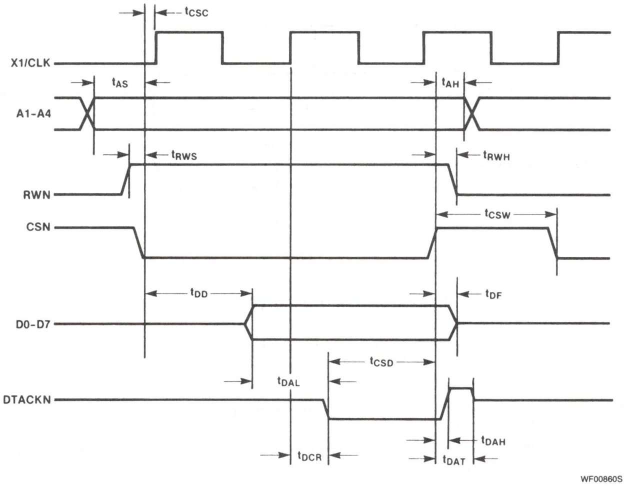

t_{CSC}
X1/CLK
t_{AS}
t_{AH}
A1-A4
t_{RWS}
t_{RWH}
RWN
t_{CSW}
CSN
t_{DD}
t_{DF}
D0-D7
t_{CSD}
t_{DAL}
DTACKN
t_{DAH}
t_{DCR}
t_{DAT}
WF00860S

Figure 2. Bus Timing (Read Cycle)

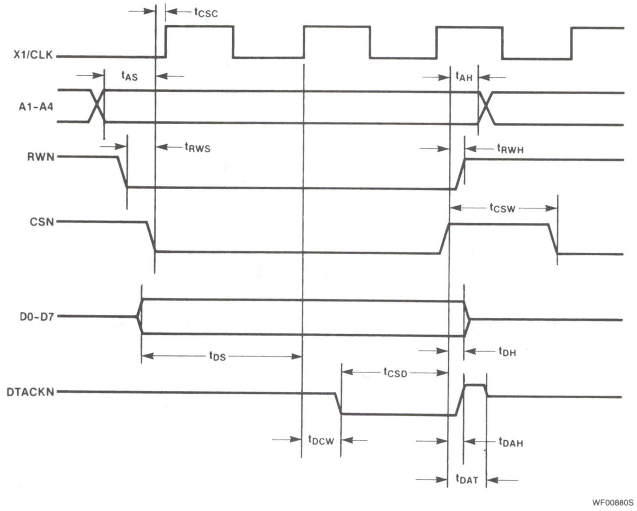

t_{CSC}
X1/CLK
t_{AS}
t_{AH}
A1-A4
RWN
t_{RWS}
t_{RWH}
CSN
t_{CSW}
D0-D7
t_{DS}
t_{DH}
t_{CSD}
DTACKN
t_{DCW}
t_{DAH}
t_{DAT}
WF00880S

Figure 3. Bus Timing (Write Cycle)

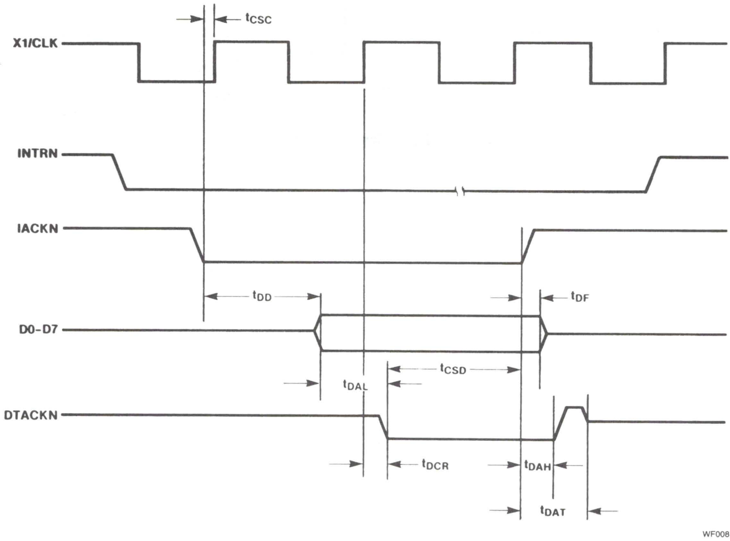

t_{CSC}
X1/CLK
INTRN
IACKN
t_{DD}
t_{DF}
D0-D7
t_{CSD}
t_{DAL}
DTACKN
t_{DCR}
t_{DAH}
t_{DAT}
WF00870S

Figure 4. Interrupt Cycle Timing

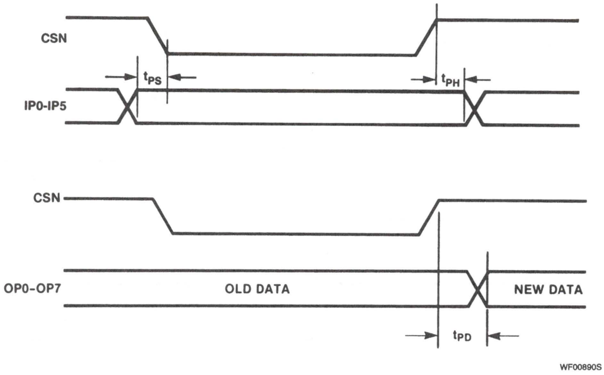
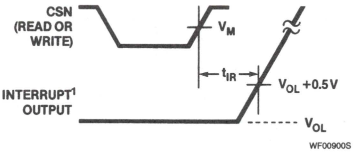
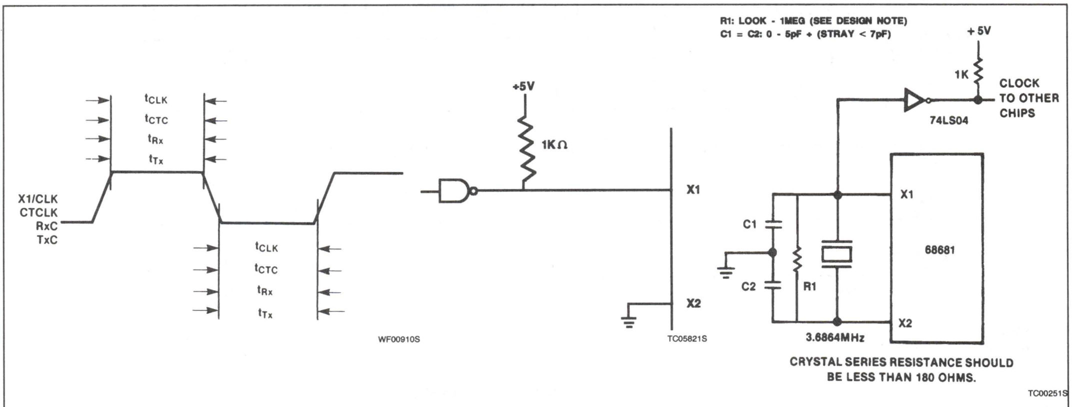

CSN
t_{PS}
t_{PH}
IP0-IP5
CSN
OP0-OP7
OLD DATA
NEW DATA
t_{PD}
WF00890S

Figure 5. Port Timing


CSN (READ OR WRITE)
V_M
t_{IR}
INTERRUPT[1] OUTPUT
V_{OL} +0.5V
V_{OL}
WF00900S

NOTES:
1. INTRN or OP3 – OP7 when used as interrupt outputs.
2. The test for open drain outputs is intended to guarantee switching of the output transistor. Measurement of this response is referenced from the midpoint of the switching signal, V_M, to a point 0.5 volts above V_{OL}. This point represents noise margin that assures true switching has occurred. Beyond this level, the effects of external circuitry and test environment are pronounced and can greatly affect the resultant measurement.

Figure 6. Interrupt Timing


t_{CLK}
t_{CTC}
t_{Rx}
t_{Tx}
X1/CLK
CTCLK
RxC
TxC
t_{CLK}
t_{CTC}
t_{Rx}
t_{Tx}
WF00910S
+5V
1KΩ
X1
X2
TC05821S
+5V
1K
CLOCK TO OTHER CHIPS
74LS04
X1
68681
C1
C2
R1
X2
3.6864MHz
TC00251S

R1: LOOK - 1MEG (SEE DESIGN NOTE)
C1 = C2: 0 - 5pF + (STRAY < 7pF)

CRYSTAL SERIES RESISTANCE SHOULD BE LESS THAN 180 OHMS.

Figure 7. Clock Timing

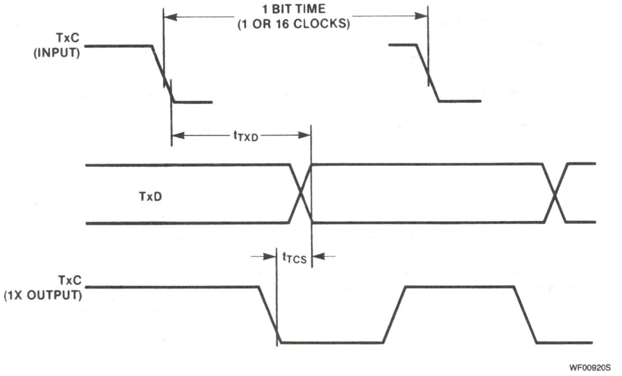

1 BIT TIME
(1 OR 16 CLOCKS)
TxC
(INPUT)
t_{TXD}
TxD
t_{TCS}
TxC
(1X OUTPUT)
WF00920S

Figure 8. Transmit Timing

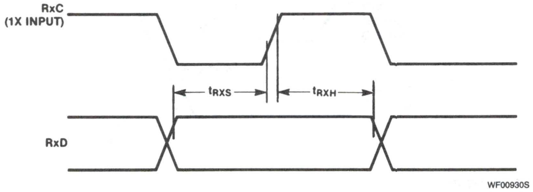
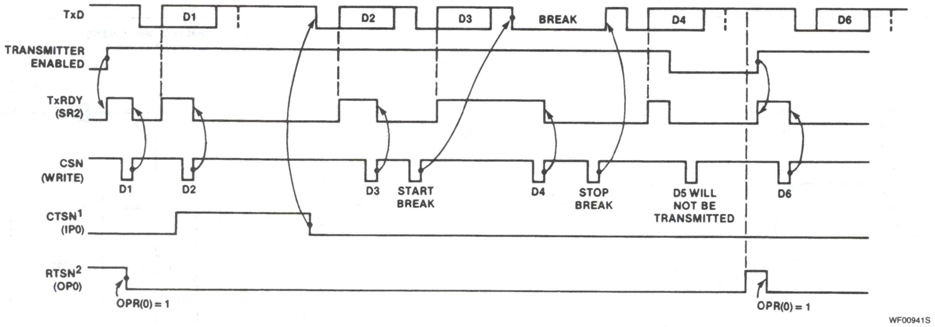
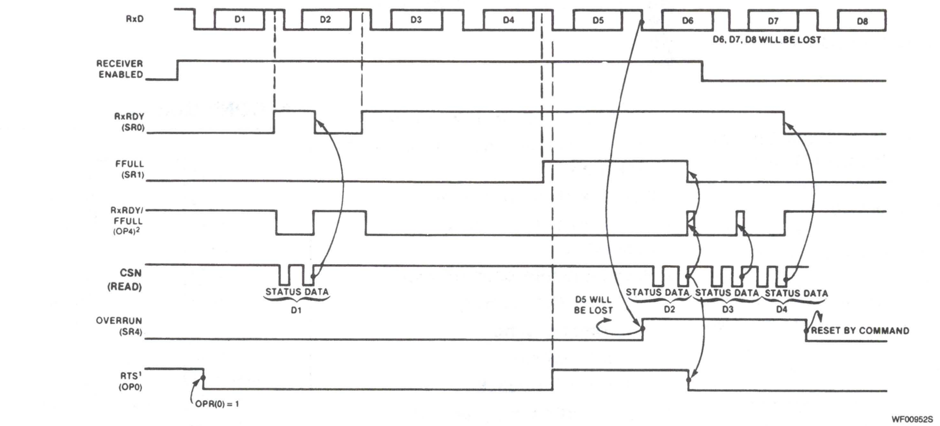

RxC
(1X INPUT)
t_{RXS}
t_{RXH}
RxD
WF00930S

Figure 9. Receive Timing


TxD
D1
D2
D3
BREAK
D4
D6
TRANSMITTER
ENABLED
TxRDY
(SR2)
CSN
(WRITE)
D1
D2
D3
START
BREAK
D4
STOP
BREAK
D5 WILL
NOT BE
TRANSMITTED
D6
CTSN1
(IP0)
RTSN2
(OP0)
OPR(0) = 1
OPR(0) = 1
WF00941S

NOTES:
1. Timing shown for MR2(4) = 1.
2. Timing shown for MR2(5) = 1.

Figure 10. Transmitter Timing


RxD
D1
D2
D3
D4
D5
D6
D7
D8
D6, D7, D8 WILL BE LOST
RECEIVER ENABLED
RxRDY (SR0)
FFULL (SR1)
RxRDY/ FFULL (OP4)²
CSN (READ)
STATUS DATA
D1
D5 WILL BE LOST
STATUS DATA
D2
STATUS DATA
D3
STATUS DATA
D4
OVERRUN (SR4)
RESET BY COMMAND
RTS¹ (OP0)
OPR(0) = 1
WF00952S

NOTES:
1. Timing shown for MR1(7) = 1.
2. Shown for OPCR(4) and MR1(6) = 0.

Figure 11. Receiver Timing

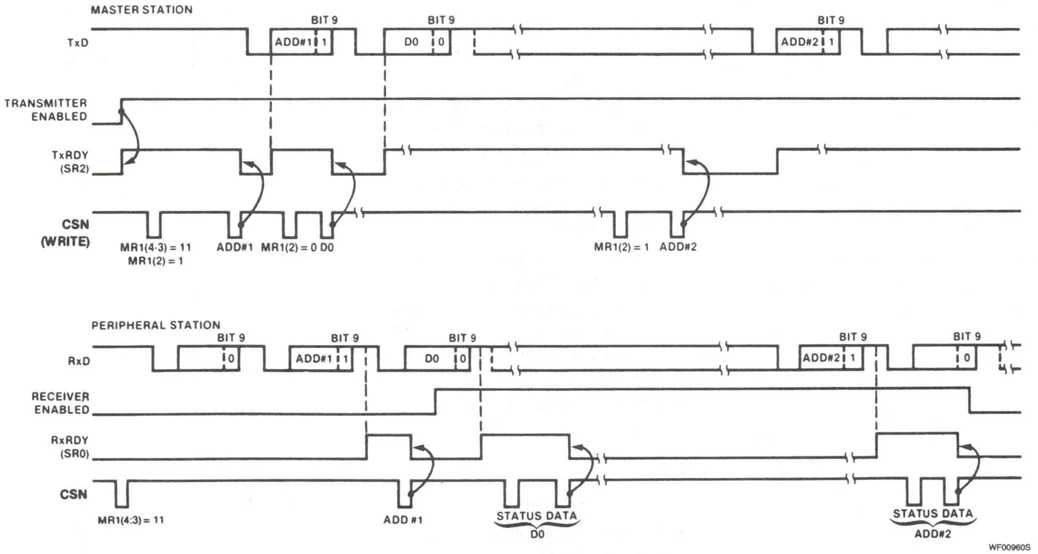

MASTER STATION
TxD
BIT 9
ADD#1
1
BIT 9
D0
0
BIT 9
ADD#2
1
TRANSMITTER ENABLED
TxRDY (SR2)
CSN (WRITE)
MR1(4-3) = 11
MR1(2) = 1
ADD#1
MR1(2) = 0
D0
MR1(2) = 1
ADD#2
PERIPHERAL STATION
RxD
BIT 9
0
BIT 9
ADD#1
1
BIT 9
D0
0
BIT 9
ADD#2
1
BIT 9
0
RECEIVER ENABLED
RxRDY (SR0)
CSN
MR1(4:3) = 11
ADD #1
STATUS DATA
D0
STATUS DATA
ADD#2
WF00960S

Figure 12. Wake-up Mode

Signetics

SCC68692

Dual Asynchronous Receiver/ Transmitter (DUART)

Preliminary Specification

Microprocessor Products

DESCRIPTION

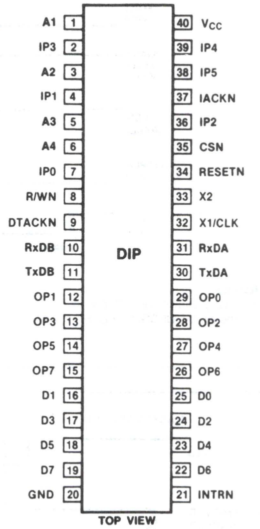
The Signetics SCC68692 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single-chip CMOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It is compatible with other S68000 family devices, and can also interface easily with other microprocessors. The DUART can be used in polled or interrupt driven systems.

The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16× clock derived from a programmable counter/timer, or an external 1× or 16× clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

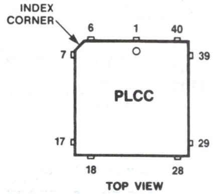
FEATURES

- **S68000 bus compatible**
- **Dual full-duplex asynchronous receiver/transmitter**
- **Quadruple buffered receiver data registers**
- **Programmable data format**
  - **5 to 8 data bits plus parity**
  - **Odd, even, no parity or force parity**
  - **1, 1.5 or 2 stop bits programmable in 1/16-bit increments**
- **Programmable baud rate for each receiver and transmitter selectable from:**
  - **18 fixed rates: 50 to 38.4k baud**
  - **One user-defined rate derived from programmable timer/counter**
  - **External 1× or 16× clock**
- **Parity, framing, and overrun error detection**
- **False start bit detection**
- **Line break detection and generation**
- **Programmable channel mode**
  - **Normal (full-duplex)**
  - **Automatic echo**
  - **Local loopback**
  - **Remote loopback**
- **Multi-function programmable 16-bit counter/timer**
- **Multi-function 6-bit input port**
  - **Can serve as clock or control inputs**
  - **Change-of-state detection on four inputs**
- **Multi-function 8-bit output port**
  - **Individual bit set/reset capability**
  - **Outputs can be programmed to be status/interrupt signals**
- **Versatile interrupt system**
  - **Single interrupt output with eight maskable interrupting conditions**
  - **Interrupt vector output on interrupt acknowledge**
  - **Output port can be configured to provide a total of up to six separate wire-ORable interrupt outputs**
- **Maximum data transfer: 1× - 1MB/sec, 16× - 125kB/sec**
- **Automatic wake-up mode for multidrop applications**
- **Start-end break interrupt/status**
- **Detects break which originates in the middle of a character**
- **On-chip crystal oscillator**
- **TTL compatible**
- **Single +5V power supply**
- **Power down mode**
- **Receiver timeout mode**

PIN CONFIGURATIONS



CD00231S



CD0044PS

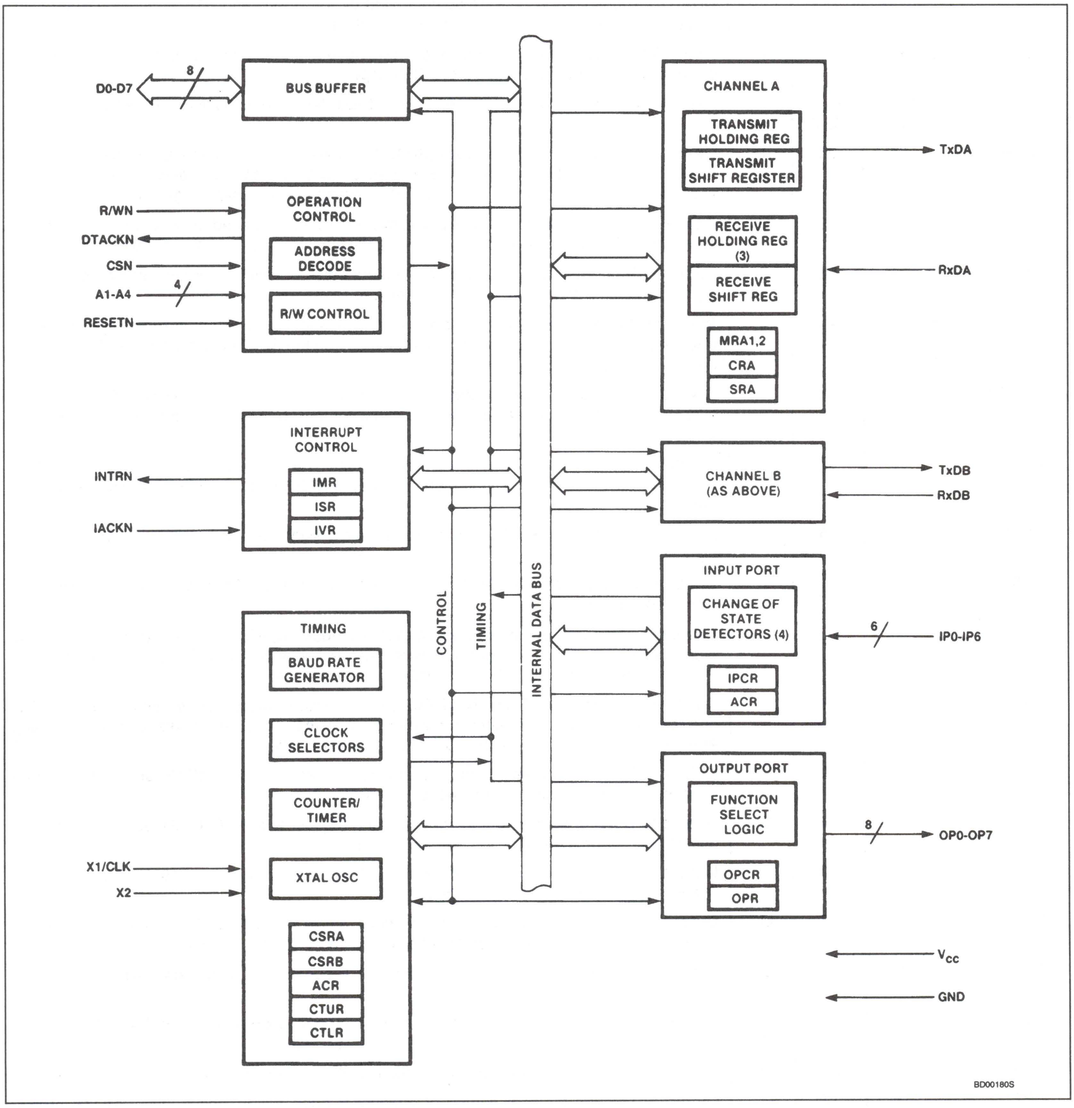
Pin	Function	Pin	Function
1	NC	23	NC
2	A1	24	INTRN
3	IP3	25	D6
4	A2	26	D4
5	IP1	27	D2
6	A3	28	D0
7	A4	29	OP6
8	IP0	30	OP4
9	R/WN	31	OP2
10	DTACKN	32	OP0
11	RxDB	33	TxDA
12	NC	34	NC
13	TxDB	35	RxDA
14	OP1	36	X1/CLK
15	OP3	37	X2
16	OP5	38	RESET
17	OP7	39	CSN
18	D1	40	IP2
19	D3	41	IACKN
20	D5	42	IP5
21	D7	43	IP4
22	GND	44	V_{CC}

Dual Asynchronous Receiver/Transmitter (DUART) SCC68692

ORDERING INFORMATION

PACKAGES	V_{CC} = +5V ±5%, T_A = 0°C to +70°C
Ceramic DIP	SCC68692C1I40
Plastic DIP	SCC68692C1N40
Plastic LCC	SCC68692CIA44

BLOCK DIAGRAM


D0-D7
8
BUS BUFFER
R/WN
DTACKN
CSN
A1-A4
4
RESETN
OPERATION CONTROL
ADDRESS DECODE
R/W CONTROL
INTRN
IACKN
INTERRUPT CONTROL
IMR
ISR
IVR
TIMING
BAUD RATE GENERATOR
CLOCK SELECTORS
COUNTER/ TIMER
X1/CLK
X2
XTAL OSC
CSRA
CSRB
ACR
CTUR
CTLR
CONTROL
TIMING
INTERNAL DATA BUS
CHANNEL A
TRANSMIT HOLDING REG
TRANSMIT SHIFT REGISTER
RECEIVE HOLDING REG (3)
RECEIVE SHIFT REG
MRA1,2
CRA
SRA
TxDA
RxDA
CHANNEL B (AS ABOVE)
TxDB
RxDB
INPUT PORT
CHANGE OF STATE DETECTORS (4)
IPCR
ACR
6
IP0-IP6
OUTPUT PORT
FUNCTION SELECT LOGIC
OPCR
OPR
8
OP0-OP7
V_{CC}
GND
BD00180S

Dual Asynchronous Receiver/Transmitter (DUART) SCC68692

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
D0 – D7	25, 16, 24 17, 23, 18 22, 19	28, 18, 27 19, 26, 20 25, 21	I/O	**Data Bus:** Bidirectional 3-State data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CSN	35	39	I	**Chip Select:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0 – D7 as controlled by the R/WN and A1 – A4 inputs. When high, places the D0 – D7 lines in the 3-State condition.
R/WN	8	9	I	**Read/Write:** A high input indicates a read cycle and a low input indicates a write cycle, when a cycle is initiated by assertion of the CSN input.
A1 – A4	1, 2, 5, 6	2, 4, 6, 7	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESETN	34	38	I	**Reset:** A low clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), initializes the IVR to hex 0F, puts OP0 – OP7 in the high state, stops the counter/timer, and puts channel A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
DTACKN	9	10	O	**Data Transfer Acknowledge:** 3-State active low output asserted in write, read, or interrupt cycles to indicate proper transfer of data between the CPU and the DUART.
INTRN	21	24	O	**Interrupt Request:** Active low, open drain output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
IACKN	37	41	I	**Interrupt Acknowledge:** Active low input indicating an interrupt acknowledge cycle. In response, the DUART will place the interrupt vector on the data bus and will assert DTACKN if it has an interrupt pending.
X1/CLK	32	36	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. If a crystal is used, a capacitor must be connected from this pin to ground (see Figure 7).
X2	33	37	I	**Crystal 2:** Connection for other side of the crystal. If a crystal is used, a capacitor must be connected from this pin to ground (see Figure 7). If X1/CLK is driven from an external source, this pin must be open.
RxDA	31	35	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. "Mark" is high, "space" is low.
RxDB	10	11	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. "Mark" is high, "space" is low.
TxDA	30	33	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the "mark" condition when the transmitter is disabled, idle, or when operating in local loopback mode. "Mark" is high, "space" is low.
TxDB	11	13	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the "mark" condition when the transmitter is disabled, idle, or when operating in local loopback mode. "Mark" is high, "space" is low.
OP0	29	32	O	**Output 0:** General purpose output, or channel A request to send (RTSAN, active low); can be deactivated automatically on receive or transmit.
OP1	12	14	O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low); can be deactivated automatically on receive or transmit.
OP2	28	40	O	**Output 2:** General purpose output, or channel A transmitter 1× or 16× clock output, or channel A receiver 1× clock output.
OP3	13	15	O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1× clock output, or channel B receiver 1× clock output.
OP4	27	30	O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.
OP5	14	16	O	**Output 5:** General purpose output or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	26	29	O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output.
OP7	15	17	O	**Output 7:** General purpose output, or channel B open drains, active low, TxRDYB output.
IP0	7	8	I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN).

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
IP1	4	5	I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN).
IP2	36	40	I	**Input 2:** General purpose input, or channel B receiver external clock input (RxCB), or counter/timer external clock input. When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP3	2	3	I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP4	39	43	I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	38	42	I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
V_{CC}	40	44	I	**Power Supply:** +5V supply input.
GND	20	22	I	**Ground**

Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

Also provided on the SCN68681 are a multipurpose 6-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

BLOCK DIAGRAM

The SCC68692 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The DTACKN output is asserted during write and read cycles to indicate to the CPU that data has been latched on a write cycle, or that valid data is present on the bus on a read cycle.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurrence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR), the interrupt status register (SR), the auxiliary control register (ACR), and the interrupt vector register (IVR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. When IACKN is asserted, and the DUART has an interrupt pending, the DUART responds by placing the contents of the IVR register on the data bus and asserting DTACKN.

Outputs OP3 – OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864nHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART. If an external signal is used instead of a crystal, X1 should be driven using a configuration similar to the one in Figure 7.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4k baud. The clock outputs from the BRG are at 16× the actual baud rate. The counter/timer can be used as a timer to produce a 16× clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection, for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Communications Channels A and B

Each communications channel of the SCC68692 comprises a full-duplex asynchronous receiver/transmitter (DUART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The input port pulse detection circuitry uses a 38.4kHz sampling clock derived from one of the baud rate generator taps. This results in a sampling period of slightly more than 25μs (assuming that the clock input is 3.6864MHz). The detection circuitry, in order to guarantee a true change in level has occurred, requires that two successive samples at the new logic level be observed. As a consequence, the minimum duration of the signal change is 25μs if the transition occurs coincident with the first sample pulse. The 50μs time refers to the situation in which the change of state is just missed and the first change of state is not detected until 25μs later.

The inputs to this unlatched 6-bit port can be read by the CPU by performing a read operation at address D16. A high input results in a logic 1 while a low input results in a logic 0. D7 will always be read as a logic 1 and D6 will reflect the level of IACKN. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs lasting longer than 25–50μs will set the corresponding bit in the input port change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice-versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E_{16} with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F16 with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The SCN68692 is conditioned to transmit data when the transmitter is enabled through the command register. The SCN68692 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted, if it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted; only if the transmitter has been disabled.

Receiver

The SCC68692 is conditioned to receive data when enabled through the command register. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16× clock for 7½ clocks (16× clock model) or at the next rising edge of the bit time clock (1× clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are "popped" thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the "character" mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the "block" mode, the status provided in the SR for these three bits is the logical-OR of the status for all characters coming to the top of the FIFO since the last "reset error" command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is "popped" only when the

Table 1. 68692 Register Addressing

A4	A3	A2	A1	READ (R/WN = 1)	WRITE (R/WN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	Rx Holding Register A (RHRA)	Tx Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	Rx Holding Register B (RHRB)	Tx Holding Register B (THRB)
1	1	0	0	Interrupt Vector Reg. (IVR)	Interrupt Vector Reg. (IVR)
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit (SP[4]) will be set upon receipt of the start bit of the new (overrunning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Timeout Mode

Under certain conditions, the user may want to set the receiver to interrupt the CPU when the receive FIFO becomes full. This can be accomplished by programming MR1[6] = 1. If a message that is only one or two characters long is received, the FIFO is not full so that ISR[1] does not set and the CPU is not interrupted. The CPU will not know that there is data in the receive FIFO. The time-out mode provides the user with a time-out interrupt via the C/T. If a character is received and the FIFO does not become full, a pre-selected period of delay can be timed out by the C/T and the CPU interrupted.

This mode is enabled by writing the appropriate command to the command register. Writing an "AX" to CRA or CRB will invoke the time-out mode for that channel. Writing a "CX" to CRA or CRB will reset the time-out mode. CTU and CTL must be loaded with a value greater than the normal receive character period. Each time a received character is transferred from the shift register to the RHR, the C/T is reloaded with the value in CTU and CTL and then restarted. If the C/T is allowed to end the count, the counter ready bit (ISR[3]) will be set. If IMR[3] is set, an interrupt will occur.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to "11" for channels A and B respectively. In this mode of operation, a "master" station transmits an address character followed by data characters for the addressed "slave" station. The slave stations, with receivers that are normally disabled, examine the received data stream and "wake-up" the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program in the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in Table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a "reset pointer" command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x switches the pointer to MR2x. the pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to Table 2 for register bit descriptions. The reserved registers at addresses H'02' and H'0A' should never be read during normal operation since they are reserved for internal diagnostics.

Table 2. Register Bit Formats

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RxRTS CONTROL	RxINT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHARACTER	
MR1A MR1B	0=no 1=yes	0=RxRDY 1=FFULL	0=char 1=block	00 = with parity 01 = force parity 10 = no parity 11 = multi-drop mode		0=even 1=odd	00=5 01=6 10=7 11=8	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	CHANNEL MODE		TxRTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A MR2B	00=Normal 01=Auto echo 10=Local loop 11=Remote loop		0=no 1=yes	0=no 1=yes	0=0.563 1=0.625 2=0.688 3=0.750	4=0.813 5=0.875 6=0.938 7=1.000	8=1.563 9=1.625 A=1.688 B=1.750	C=1.813 D=1.875 E=1.938 F=2.000

NOTE:
*Add 0.5 to values shown for 0–7 if channel is programmed for 5 bits/char.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
CSRA CSRB	See Text				See Text			

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	MISCELLANEOUS COMMANDS				DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
CRA CRB	See Text				0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA SRB	0=no 1=yes *	0=no 1=yes *	0=no 1=yes *	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes

NOTE:
*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits 4:0. These bits are cleared by a "reset error status" command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1=TxRDYB	0 = OPR[6] 1=TxRDYA	0 = OPR[5] 1=RxRDY/ FFULLB	0 = OPR[4] 1=RxRDY/ FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB(1 ×) 11 = RxCB(1 ×)		00 = OPR[2] 01 = TxCA(16 ×) 10 = TxCA(1 ×) 11 = RxCA(1 ×)	

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See Table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

Table 2. Register Bit Formats (Continued)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IPCR	**ΔIP3**	**ΔIP2**	**ΔIP1**	**ΔIP0**	**IP3**	**IP2**	**IP1**	**IP0**
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ISR	**INPUT PORT CHANGE**	**Δ BREAK B**	**RxRDY/ FFULLB**	**TxRDYB**	**COUNTER READY**	**Δ BREAK A**	**RxRDY/ FFULLA**	**TxRDYA**
	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes	0=no 1=yes

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IMR	**IN. PORT CHANGE INT**	**Δ BREAK B INT**	**RxRDY/ FFULLB INT**	**TxRDYB INT**	**COUNTER READY INT**	**Δ BREAK A INT**	**RxRDY/ FFULLA INT**	**TxRDYA INT**
	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on	0=off 1=on

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTUR	**C/T[15]**	**C/T[14]**	**C/T[13]**	**C/T[12]**	**C/T[11]**	**C/T[10]**	**C/T[9]**	**C/T[8]**

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CTLR	**C/T[7]**	**C/T[6]**	**C/T[5]**	**C/T[4]**	**C/T[3]**	**C/T[2]**	**C/T[1]**	**C/T[0]**

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
IVR	**IVR[7]**	**IVR[6]**	**IVR[5]**	**IVR[4]**	**IVR[3]**	**IVR[2]**	**IVR[1]**	**IVR[0]**

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a "set pointer" command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control

This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select

This bit selects either the channel A receiver ready status (RxRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the "character" mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the "block" mode, the status provided in the SR for these bits is the accumulation (logical-OR) of the status for all characters coming to the top of the FIFO since the last "reset error" command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select

If "with parity" or "force parity" is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select

This bit selects the parity type (odd or even) if the "with parity" mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the "force parity" mode is programmed. It has no effect if the "no parity" mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits per Character Select
This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select
Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control
This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Verify the message is sent by waiting until the transmit ready status (TxRDY) is asserted. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-Send Control
If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select
This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1 9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1 1/16 to 2 stop bits can be programmed in increments of 1/16 bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1× clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select
This field selects the baud rate clock for the channel A receiver as follows:

	Baud Rate Clock = 3.6864MHz	
CSRA[7:4]	**ACR[7] = 0**	**ACR[7] = 1**
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4k	19.2k
1 1 0 1	Timer	Timer
1 1 1 0	IP4 — 16×	IP4 — 16×
1 1 1 1	IP4 — 1×	IP4 — 1×

The receiver clock is always a 16 × clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select

This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRA[3:0]	**ACR[7] = 0**	**ACR[7] = 1**
1 1 1 0	IP3 — 16×	IP3 — 16×
1 1 1 1	IP3 — 1×	IP3 — 1×

The transmitter clock is always a 16× clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select

This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[7:4]	**ACR[7] = 0**	**ACR[7] = 1**
1 1 1 0	IP2 — 16×	IP2 — 16×
1 1 1 1	IP2 — 1×	IP2 — 1×

The receiver clock is always a 16× clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select

This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[3:0]	**ACR[7] = 0**	**ACR[7] = 1**
1 1 1 0	IP5 — 16×	IP5 — 16×
1 1 1 1	IP5 — 1×	IP5 — 1×

The transmitter clock is always a 16× clock except for CSRB[3:0] = 1111

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting; e.g., the "enable transmitter" and "reset transmitter" commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands

The encoded value of this field may be used to specify a single command as follows:

CRA[7:4]	COMMAND
0 0 0 0	No command.
0 0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.
0 0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
0 1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
0 1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
0 1 1 0	Start break. Forces the TxDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
0 1 1 1	Stop Break. The TxDA line will go high (marking) within two bit times. TxDA will remain high for one bit time before the next character, if any, is transmitted.
1 0 0 0	Assert RTSN. Causes the RTSN output to be asserted (low).
1 0 0 1	Negate RTSN. Causes the RTSN output to be negated (high).
1 0 1 0	Set special time-out mode on with this channel as the channel to restart the C/T as each receive character is transferred from the serial shift register to the RHR.
1 0 1 1	Not used.
1 1 0 0	Reset special time-out mode.
1 1 0 1	Not used.
1 1 1 0	Power Down mode on. In this mode, the DUART oscillator is stopped and all functions requiring this clock are suspended. The contents of all registers are saved. It is recommended that the transmitter and receiver be disabled prior to placing the DUART in this mode. This bit is reset with the assertion of RESET. This command is in CRA only. **Design Note:** The part will not output DTACKN while in Power Down mode, use automatic DTACKN generation.
1 1 1 1	Power Down Mode Normal Run. This command resets the Power Down mode. This command is in CRA only.

CRA[3] — Disable Channel A Transmitter

This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter

Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver

This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver

Enables operation of the channel A receiver. If not in the special wake-up mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting; e.g., the "enable transmitter" and "reset transmitter" commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA with the exceptions of the last two commands "EX" and "FX" which are used for power down mode. These two commands are not used in CRB. All other control actions that apply to CRA also apply to CRB.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break
This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received: further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1× clock).

When this bit is set, the channel A "change in break" bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error
This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error
This bit is set when the "with parity" or "force parity" mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error
This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a "reset error status" command.

SRA[3] — Channel A Transmitter Empty (TxEMTA)
This bit will be set when the channel A transmitter underruns; i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA)
This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

SRA[1] — Channel A FIFO Full (FFULLA)
This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full; i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full. FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA)
This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, only if (after this read) there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select
This bit programs the OP7 output to provide one of following:
- The complement of OPR[7]
- The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select
This bit programs the OP6 output to provide one of the following:
- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select
This bit programs the OP5 output to provide one of the following:
- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select
This bit programs the OP4 output to provide one of the following:
- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open-drain output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select
This field programs the OP3 output to provide one of the following:
- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open-drain output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1× clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1× clock is output.
- The 1× clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1× clock is output.

OPCR[1:0] — OP2 Output Select
This field programs the OP2 output to provide one of the following:
- The complement of OPR[2]
- The 16× clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be 1× clock if CSRA[3:0] = 1111.
- The 1× clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1× clock is output.
- The 1× clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1× clock is output.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select
This bit selects one of two sets of baud rates to be generated by the BRG:

Table 3. Baud Rate Generator Characteristics Crystal or Clock = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16X CLOCK (KHz)	ERROR (PERCENT)
50	0.8	0
75	1.2	0
110	1.759	−0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	−0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2k	307.2	0
38.4k	614.4	0

NOTE:
Duty cycle of 16× clock is 50% ±1%

Table 4. ACR [6:4] Field Definition

ACR [6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)[1]
0 0 1	Counter	TxCA — 1× clock of channel A transmitter
0 1 0	Counter	TxCB — 1× clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)[1]
1 0 1	Timer	External (IP2) divided by 16[1]
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

NOTE:
[1] In these modes, the channel B receiver clock should normally be generated from the baud rate generator.

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05k, 1.2k, 2.4k, 4.8k, 7.2k, 9.6k, and 38.4k baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k, and 19.2k baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in Table 3.

ACR[6:4] — Counter/Timer Mode and Clock Source Select
This field selects the operating mode of the counter/timer and its clock source as shown in Table 4.

ACR[3:0] — IP3, IP2, IP1, IP0 Change of State Interrupt Enable
This field selects which bits of the Input Port Change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the "on" state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the "off" state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change of State
These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State
These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a "1" and the corresponding bit in the IMR is also a "1", the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initalized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status
This bit is a "1" when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

ISR[6] — Channel B Change in Break
This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B "reset break change interrupt" command.

ISR[5] — Channel B Receiver Ready or FIFO Full
The function of this bit is programmed by MR1B[6]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is "popped". If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full; i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[4] — Channel B Transmitter Ready
This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready
In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change in Break
This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A "reset break change interrupt" command.

ISR[1] — Channel A Receiver Ready or FIFO Full
The function of this bit is programmed by MR1A[6]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is "popped". If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full; i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready
This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a "1" and the corresponding bit in the IMR is also a "1", the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3–OP7 or the reading of the ISR.

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs respectively of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 0002_{16}. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A4–A1 = 1110) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR. The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A4–A1 = 1111). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

On power-up and after reset, the timer/counter runs in timer mode and can only be restarted. Because it cannot be shut off or stopped, and runs continuously in timer mode, it is recommended that at initialization, the output port, OP3, should be masked off through the OPCR[3:2] = 00 until the T/C is programmed to the desired operational state.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8 bits to the upper 8 bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

IVR — Interrupt Vector Register

This register contains the interrupt vector. The register is initialized to H'0F' by RESET. The contents of the register are placed on the data bus when the DUART responds to a valid interrupt acknowledge cycle.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +6.0	V

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2			V
V_{IH}	Input high voltage (X1/CLK)		$0.9V_{CC}$		V_{CC}	V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.d. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	µA
I_{LL}	Data bus 3-State leakage current	$V_O = 0.4$ to V_{CC}	−10		10	µA
I_{X1L}	X1/CLK low input current	$V_{IN} = 0$, X2 floated	−100	−30	0	µA
I_{X1H}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 floated	0	+30	100	µA
I_{Od}	Open-drain output leakage current	$V_O = 0.4$ to V_{CC}	−10		10	µA
I_{CC}	Power supply current				150	mA
When oscillator is in power-down mode:						
I_{X1H}	X1/CLK high input current	$V_{IN} = V_{CC}$, X2 floated	2	6	10	mA
I_{CC}	Power supply current			0.8	4	mA
	Standby				500	µA

NOTES:

4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all inputs except X1/CLK swing between 0.4V and 2.4V with a transition time of 20ns maximum. For X1/CLK, this swing is between 0.4V and 4.4V. All time measurements are referenced at input voltages of 0.8V and 2V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. Test condition for outputs: $C_L = 150pF$, except interrupt outputs. Test condition for interrupt outputs: $C_L = 50pF$, $R_L = 2.7k\Omega$ to V_{CC}.
8. This specification will impose maximum 68000 CPU CLK to 6MHz. Higher CPU CLK can be used if repeating bus reads are not performed. Consecutive write operations to the same command register require at least three edges of the X1 clock between writes.
9. This specification imposes a lower bound on CSN and IACKN low, guaranteeing that it will be low for at least 1 CLK period. This requirement is made on CSN only to insure assertion of DTACKN and not to guarantee operation of the part.
10. This specification is made only to insure that DTACKN is asserted with respect to the rising edge of the X1/CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the setup time is violated, DTACKN may be asserted as shown, or may be asserted one clock cycle later.

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%[4, 5, 6, 7]

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Typ	Max	
Reset timing (Figure 1)					
t_{RES}	RESETN pulse width	1			μs
Bus timing (Figures 2, 3, 4)					
t_{AS}	A1–A4 setup time to CSN low	10			ns
t_{AH}	A1–A4 hold time from CSN high	0			ns
t_{RWS}	RWN setup time to CSN high	0			ns
t_{RWH}	RWN holdup time to CSN high	0			ns
t_{CSW}[8]	CSN high pulse width	160			ns
t_{CSD}[9]	CSN or IACKN high from DTACKN low	20			ns
t_{DD}	Data valid from CSN or IACKN low			175	ns
t_{DF}	Data bus floating from CSN or IACKN high			65	ns
t_{DS}	Data setup time to CLK high	100			ns
t_{DH}	Data hold time from CSN high	0			ns
t_{DAL}	DTACKN low from read data valid	0			ns
t_{DCR}	DTACKN low (read cycle) from CLK high			125	ns
t_{DCW}	DTACKN low (write cycle) from CLK high			125	ns
t_{DAH}	DTACKN high from CSN or IACKN high			100	ns
t_{DAT}	DTACKN high impedance from CSN or IACKN high			125	ns
t_{CSC}[10]	CSN or IACKN setup time to clock high	90			ns
Port timing (Figure 5)					
t_{PS}	Port input setup time to CSN low	0			ns
t_{PH}	Port input hold time from CSN high	0			ns
t_{PD}	Port output valid from CSN high			370	ns
Interrupt reset timing (Figure 6)					
t_{IR}	INTRN, or OP3–OP7 when used as interrupts, negated from:				
	Read RHR (RxRDY/FFULL interrupt)			370	ns
	Write THR (TxRDY interrupt)			370	ns
	Reset command (delta break interrupt)			370	ns
	Stop C/T command (counter interrupt)			370	ns
	Read IPCR (input port change interrupt)			370	ns
	Write IMR (clear of interrupt mask bit)			270	ns
Clock timing (Figure 7)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2	3.6864	4	MHz
t_{CTC}	CTCLK high or low time	100			ns
f_{CTC}	CTCLK frequency	100		4M	Hz
t_{RX}	RxC high or low time	220			ns
f_{RX}	RxC frequency (16×)	100		2M	Hz
	(1×)	100		1M	Hz
t_{TX}	TxC high or low time	220			ns
f_{TX}	TxC frequency (16×)	0		2	MHz
	(1×)	0		1	MHz
Transmitter timing (Figure 8)					
t_{TXD}	TxD output delay from TxC low			350	ns
t_{TCS}	Output delay from TxC low to TxD data output			150	ns
Receiver timing (Figure 9)					
t_{RXS}	RxD data setup time to RxC high	240			ns
t_{RXH}	RxD data hold time from RxC high	200			ns

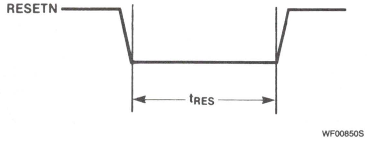

RESETN
t_{RES}
WF00850S

Figure 1. Reset Timing

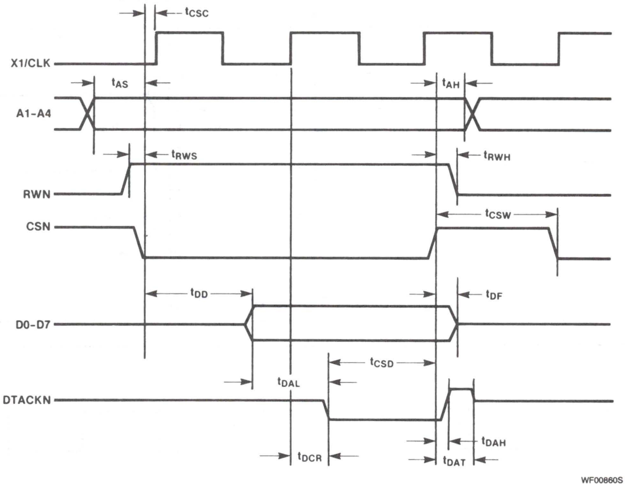

t_{CSC}
X1/CLK
t_{AS}
t_{AH}
A1-A4
t_{RWS}
t_{RWH}
RWN
CSN
t_{CSW}
t_{DD}
t_{DF}
D0-D7
t_{CSD}
t_{DAL}
DTACKN
t_{DAH}
t_{DCR}
t_{DAT}
WF00860S

Figure 2. Bus Timing (Read Cycle)

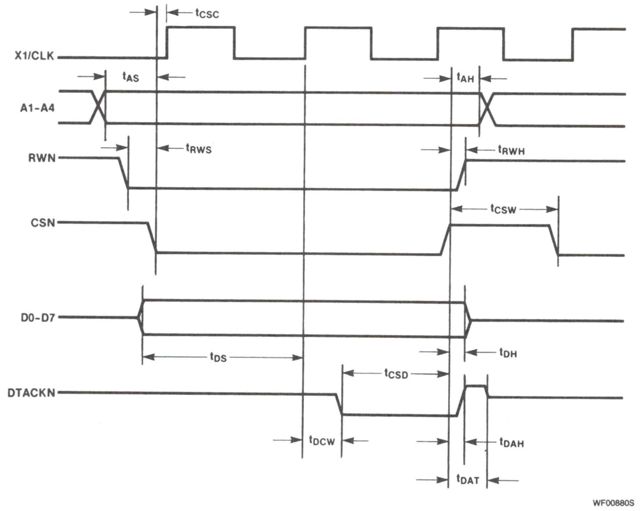

t_{CSC}
X1/CLK
t_{AS}
t_{AH}
A1-A4
RWN
t_{RWS}
t_{RWH}
CSN
t_{CSW}
D0-D7
t_{DS}
t_{DH}
t_{CSD}
DTACKN
t_{DCW}
t_{DAH}
t_{DAT}
WF00880S

Figure 3. Bus Timing (Write Cycle)

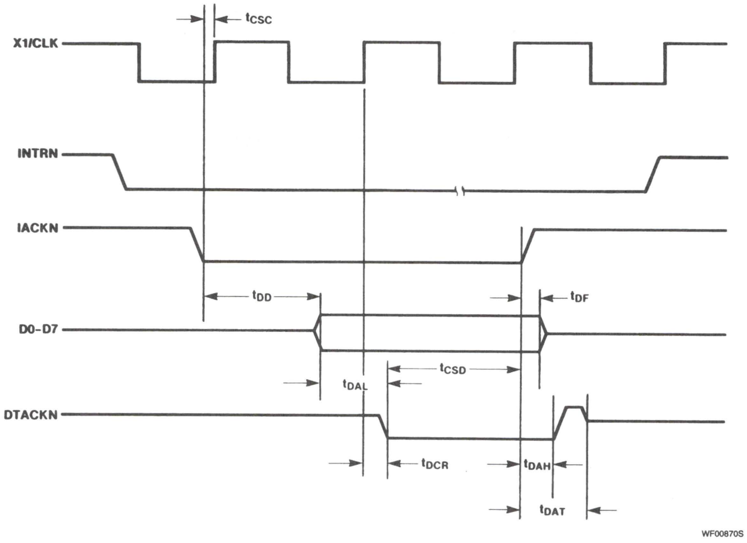

t_{CSC}
X1/CLK
INTRN
IACKN
t_{DD}
t_{DF}
D0-D7
t_{CSD}
t_{DAL}
DTACKN
t_{DCR}
t_{DAH}
t_{DAT}
WF00870S

Figure 4. Interrupt Cycle Timing

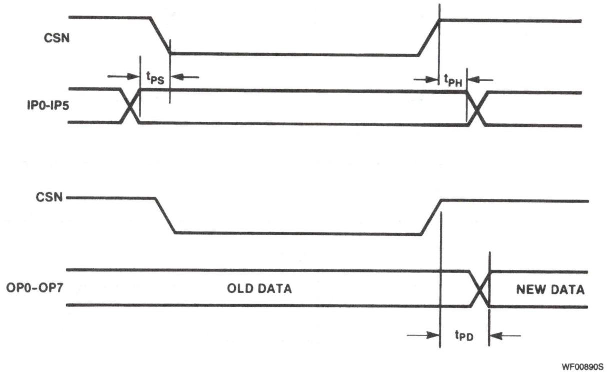
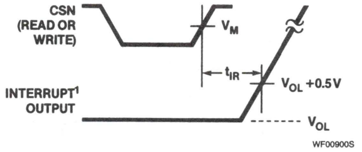

CSN
tPS
tPH
IP0-IP5
CSN
OP0-OP7
OLD DATA
NEW DATA
tPD
WF00890S

Figure 5. Port Timing


CSN
(READ OR
WRITE)
VM
tIR
VOL +0.5V
INTERRUPT1
OUTPUT
VOL
WF00900S

NOTES:

1. INTRN or OP3 – OP7 when used as interrupt outputs.
2. The test for open drain outputs is intended to guarantee switching of the output transistor. Measurement of this response is referenced from the midpoint of the switching signal, V_M, to a point 0.5 volts above V_{OL}. This point represents noise margin that assures true switching has occurred. Beyond this level, the effects of external circuitry and test environment are pronounced and can greatly affect the resultant measurement.

Figure 6. Interrupt Timing

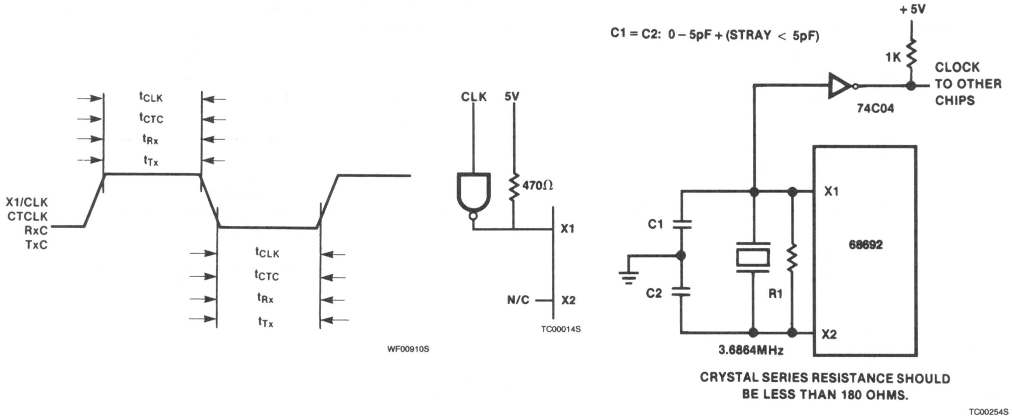

tCLK
tCTC
tRx
tTx
X1/CLK
CTCLK
RxC
TxC
tCLK
tCTC
tRx
tTx
WF00910S
CLK
5V
470Ω
X1
N/C
X2
TC00014S
C1 = C2: 0 – 5pF + (STRAY < 5pF)
+5V
1K
CLOCK
TO OTHER
CHIPS
74C04
X1
C1
68692
C2
R1
X2
3.6864MHz
CRYSTAL SERIES RESISTANCE SHOULD
BE LESS THAN 180 OHMS.
TC00254S

Figure 7. Clock Timing

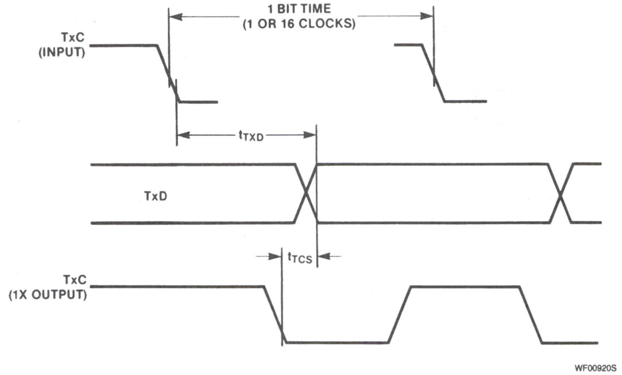

1 BIT TIME
(1 OR 16 CLOCKS)
TxC
(INPUT)
t_{TXD}
TxD
t_{TCS}
TxC
(1X OUTPUT)
WF00920S

Figure 8. Transmit Timing

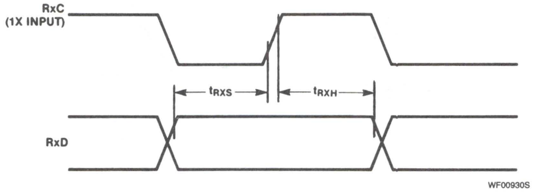
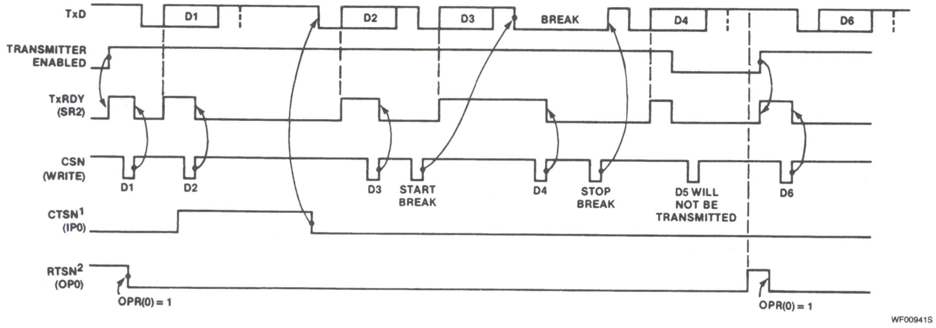
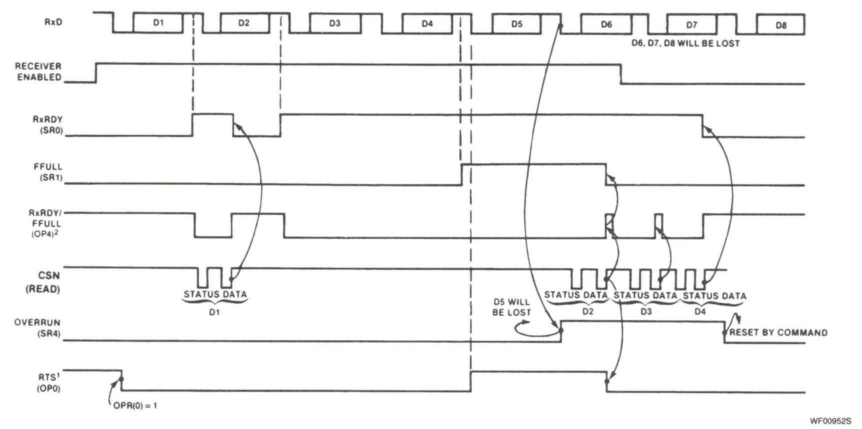

RxC
(1X INPUT)
t_{RXS}
t_{RXH}
RxD
WF00930S

Figure 9. Receive Timing


TxD
D1
D2
D3
BREAK
D4
D6
TRANSMITTER
ENABLED
TxRDY
(SR2)
CSN
(WRITE)
D1
D2
D3
START
BREAK
D4
STOP
BREAK
D5 WILL
NOT BE
TRANSMITTED
D6
CTSN¹
(IP0)
RTSN²
(OP0)
OPR(0) = 1
OPR(0) = 1
WF00941S

NOTES:
1. Timing shown for MR2(4) = 1.
2. Timing shown for MR2(5) = 1.

Figure 10. Transmitter Timing


RxD
D1
D2
D3
D4
D5
D6
D7
D8
D6, D7, D8 WILL BE LOST
RECEIVER ENABLED
RxRDY (SR0)
FFULL (SR1)
RxRDY/ FFULL (OP4)2
CSN (READ)
STATUS DATA
D1
D5 WILL BE LOST
STATUS DATA
D2
STATUS DATA
D3
STATUS DATA
D4
OVERRUN (SR4)
RESET BY COMMAND
RTS1 (OP0)
OPR(0) = 1
WF00952S

NOTES:

1. Timing shown for MR1(7) = 1.
2. Shown for OPCR(4) and MR1(6) = 0.

Figure 11. Receiver Timing

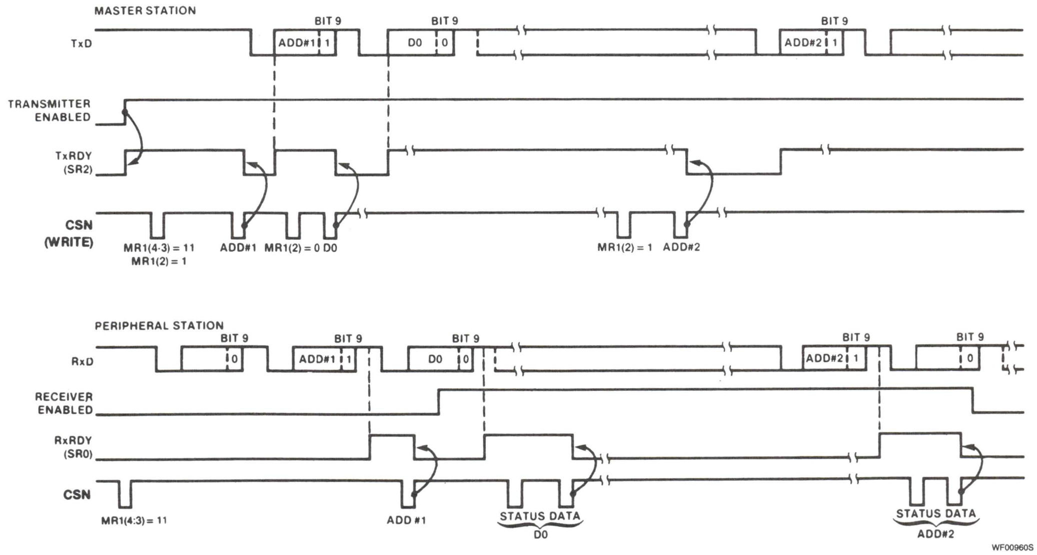

MASTER STATION
BIT 9
BIT 9
BIT 9
TxD
ADD#1 1
D0 0
ADD#2 1
TRANSMITTER ENABLED
TxRDY (SR2)
CSN (WRITE)
MR1(4·3) = 11
MR1(2) = 1
ADD#1
MR1(2) = 0
D0
MR1(2) = 1
ADD#2
PERIPHERAL STATION
BIT 9
BIT 9
BIT 9
BIT 9
BIT 9
RxD
0
ADD#1 1
D0 0
ADD#2 1
0
RECEIVER ENABLED
RxRDY (SR0)
CSN
MR1(4:3) = 11
ADD #1
STATUS DATA
D0
STATUS DATA
ADD#2
WF00960S

Figure 12. Wake-up Mode

Section 3
Microcontroller Products

INDEX

SCN8049 Series	Single-Chip 8-Bit Microcontroller	3-3
SCN8031AH/SCN8051AH	Single-Chip 8-Bit Microcontroller	3-18
SCN8032AH/SCN8052AH	Single-Chip 8-Bit Microcontroller	3-29
SC80C31B/SC80C51B	CMOS Single-Chip 8-Bit Microcontroller	3-40
SC80C451/SC83C451	I/O Expanded CMOS Microcontroller	3-54
SC87C451	CMOS Single-Chip 8-Bit EPROM Microcontroller	3-70
SC83C751	CMOS Single-Chip 8-Bit Microcontroller	3-71
SC80C552/SC83C552	CMOS Automotive and Industrial Microcontroller	3-73
SC80C652/SC83C652	CMOS Single-Chip 8-Bit Microcontroller	3-76
SC96AH Series	Single-Chip 16-Bit Microcontrollers	3-79
SC96BH Series	Single-Chip 16-Bit Microcontrollers	3-103
PCB5010/11	Single-Chip Digital Signal Processors	3-106
I^2C Bus Specification		3-111
AN408	SC83C451 Operation of Port 6	3-123
8X305	Microcontroller	3-134
8X310	Interrupt Control Coprocessor	3-159
8X320	Bus Interface Register Array	3-172
8X330	Floppy Disk Formatter/Controller	3-180
8X353	Bipolar RAM (32 × 8)	3-197
8X355	LIFO Stack Memory (32 × 8)	3-205
8X360	Memory Address Director (MAD)	3-214
8X371	8-Bit Latched Bidirectional I/O Port	3-223
8X372/8X376	Addressable/Bidirectional I/O Port	3-230
8X374	Addressable/Bidirectional I/O Port with Parity	3-239
8X382	4-Input/4-Output Addressable I/O Port	3-249
8X401	Microcontroller	3-259
AN4011	An 8X401-Based Raster Graphics System	3-279
8X411	Priority Interrupt Encoder	3-304
8X450	RAM	3-306
8X470	I/O Port	3-313
MCCAP 8X300/8X305	Cross Assembler Program	3-327
FORTRESS 8X401	Cross Assembler Program	3-328
8X300 KT1SK	8X305 Prototyping System	3-329
8X305 Development System	American Automation EZ-PRO	3-331
I/O Port Programmer		3-332

SCN8049 Series
SCN8049, SCN8050, SCN8039, SCN8040 Single-Chip 8-Bit Microcontroller

Microprocessor Products

Product Specification

DESCRIPTION

The Signetics SCN8049 Series microcontrollers are self-contained, 8-bit processors which contain the system timing, control logic, RAM data memory, ROM program memory (8048/49/50 only), and I/O lines necessary to implement dedicated control functions. All SCN8049 Series devices are pin and program compatible, differing only in the size of the on-board program ROM and data RAM, as follows:

TYPE	RAM SIZE	ROM SIZE
SCN8049	128 × 8	2K × 8
SCN8050	256 × 8	4K × 8
SCN8039	128 × 8	—
SCN8040	256 × 8	—

Program memory can be expanded externally up to a maximum total of 4K bytes without paging. Data memory can also be expanded externally. I/O capabilities can be expanded using standard devices or the 8243 I/O expander.

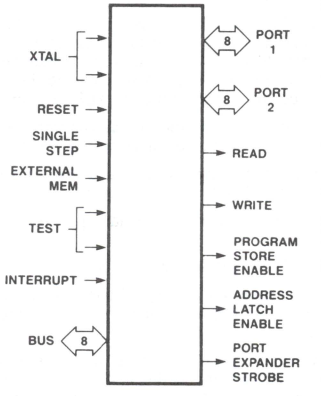
The SCN8049 Series processors are designed to be efficient control processors as well as arithmetic processors. They provide an instruction set which allows the user to directly set and reset individual lines within its I/O ports as well as test individual bits within the accumulator. A large variety of branch and table look-up instructions make these processors very efficient in implementing standard logic functions. Also, special attention has been given to code efficiency. Over 70% of the instructions are a single byte long and all others are only 2 bytes long.

An on-chip 8-bit counter is provided which can count, under program control, either internal clock pulses (with a divide by 32 prescaler) or external events. The counter can be programmed to cause an interrupt on terminal count.

FEATURES

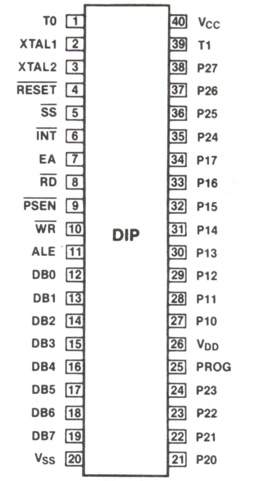
- **8-bit CPU, ROM, RAM, I/O in a 40-pin package**
- **24 quasi bidirectional I/O lines**
- **Two test inputs**
- **Internal counter/timer**
- **Single-level vectored interrupts: external, counter/timer**
- **Over 90 instructions, 70% single byte**
- **1.36μs or 2.5μs instruction cycle, all instructions one or two cycles**
- **Expandable memory and I/O**
- **Low voltage standby**
- **TTL compatible inputs and outputs**
- **Single +5V power supply**

LOGIC SYMBOL


XTAL
RESET
SINGLE STEP
EXTERNAL MEM
TEST
INTERRUPT
BUS
8
PORT 1
8
PORT 2
8
READ
WRITE
PROGRAM STORE ENABLE
ADDRESS LATCH ENABLE
PORT EXPANDER STROBE

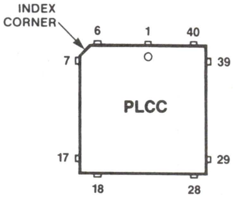
LS00350S

PIN CONFIGURATIONS


T0 1
XTAL1 2
XTAL2 3
RESET 4
SS 5
INT 6
EA 7
RD 8
PSEN 9
WR 10
ALE 11
DB0 12
DB1 13
DB2 14
DB3 15
DB4 16
DB5 17
DB6 18
DB7 19
Vss 20
DIP
40 Vcc
39 T1
38 P27
37 P26
36 P25
35 P24
34 P17
33 P16
32 P15
31 P14
30 P13
29 P12
28 P11
27 P10
26 VDD
25 PROG
24 P23
23 P22
22 P21
21 P20

TOP VIEW

CD00190S


INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28

TOP VIEW

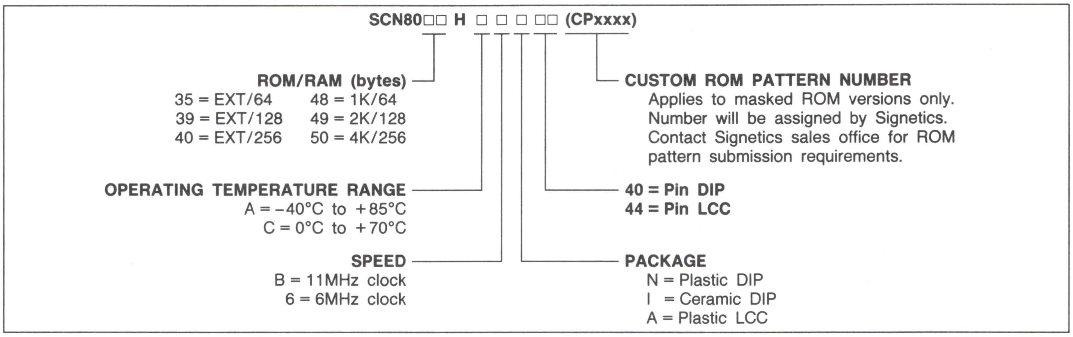
CD0044PS

Pin	Function	Pin	Function
1	NC	23	NC
2	T0	24	P20
3	XTAL1	25	P21
4	XTAL2	26	P22
5	RESET	27	P23
6	SS	28	PROG
7	INT	29	V_{DD}
8	EA	30	P10
9	RD	31	P11
10	PSEN	32	P12
11	WR	33	P13
12	NC	34	NC
13	ALE	35	P14
14	DB0	36	P15
15	DB1	37	P16
16	DB2	38	P17
17	DB3	39	P24
18	DB4	40	P25
19	DB5	41	P26
20	DB6	42	P27
21	DB7	43	T1
22	V_{SS}	44	V_{CC}

SCN8049, SCN8050, SCN8039, SCN8040 Single-Chip 8-Bit Microcontroller

SCN8049 Series

ORDERING INFORMATION



SCN80□□ H □ □ □ □□ (CPxxxx)

ROM/RAM (bytes)
35 = EXT/64 48 = 1K/64
39 = EXT/128 49 = 2K/128
40 = EXT/256 50 = 4K/256

OPERATING TEMPERATURE RANGE
A = −40°C to +85°C
C = 0°C to +70°C

SPEED
B = 11MHz clock
6 = 6MHz clock

CUSTOM ROM PATTERN NUMBER
Applies to masked ROM versions only.
Number will be assigned by Signetics.
Contact Signetics sales office for ROM pattern submission requirements.

40 = Pin DIP
44 = Pin LCC

PACKAGE
N = Plastic DIP
I = Ceramic DIP
A = Plastic LCC

PIN DESCRIPTION

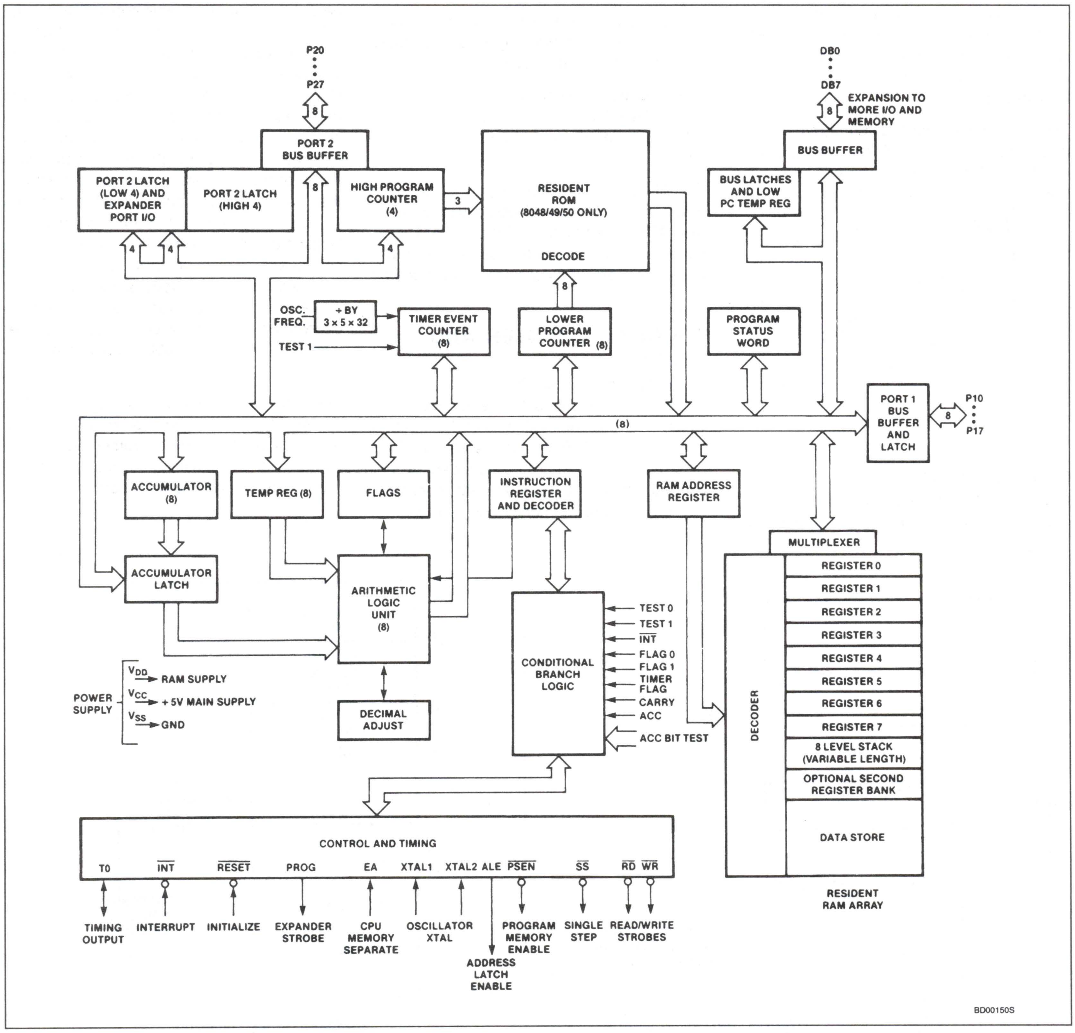
MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
V_{SS}	20	22		Circuit ground potential.
V_{DD}	26	29		Low power standby.
V_{CC}	40	44		**Main Power Supply:** +5V during operation.
PROG	25	28	O	Output strobe for 8243 I/O expander.
P10 – P17	27 – 34	30 – 33, 35 – 38	I/O	**Port 1:** 8-bit quasi-bidirectional port.
P20 – P27	21 – 24, 35 – 38	24 – 27, 39 – 42	I/O	**Port 2:** 8-bit quasi-bidirectional port. P20-23 contain the four high-order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
DB0 – DB7	12 – 19	14 – 21	I/O	**Data Bus:** True bidirectional port which can be written or read synchronously using the $\overline{\text{RD}}$, $\overline{\text{WR}}$ strobes. The port can also be statically latched. Contains the eight low-order program counter bits during an external program memory fetch and receives the addressed instruction under the control of $\overline{\text{PSEN}}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{\text{RD}}$ and $\overline{\text{WR}}$.
T0	1	2	I	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 and be designated as a clock output using the ENT0 CLK instruction.
T1	39	43	I	Input pin testable using the JT1 and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
XTAL1	2	3	I	**Crystal 1:** One side of the crystal input for internal oscillator. Also input for external source (non-TTL V_{IH}).
XTAL2	3	4	I	**Crystal 2:** Other side of crystal input.
$\overline{\text{INT}}$	6	7	I	**Interrupt:** Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. Interrupt must remain low for at least three machine cycles for proper operation.
$\overline{\text{RESET}}$	4	5	I	**Reset:** Used to initialize the microcomputer. Active low. Internal pullup ~75KΩ. During program verification the address is latched by a "0" to "1" transition on $\overline{\text{RESET}}$ and the data at the addressed location is output on BUS.
$\overline{\text{RD}}$	8	9	O	**Read:** Output strobe activated during a bus read. Can be used to enable data onto the bus from an external device. Used as a read strobe to external data memory.
$\overline{\text{WR}}$	10	11	O	**Write:** Output strobe during a bus write. Used as write strobe to external data memory.
ALE	11	13	O	**Address Latch Enable:** Occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{\text{PSEN}}$	9	10	O	**Program Store Enable:** Output occurs only during a fetch to external program memory.
$\overline{\text{SS}}$	5	6	I	**Single Step:** Can be used in conjunction with ALE to "single step" the processor through each instruction.
EA	7	8	I	**External Access:** Forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification.

NOTE:
Each pin on these ports can be assigned, under program control, to be an input or an output. A pin is designated as an input by writing a logic "1" to the pin. $\overline{\text{RESET}}$ sets all pins to the input mode. Each pin has an internal pullup of approximately 50kΩ.

FUNCTIONAL DESCRIPTION

The following is a general functional description of the SCN8049 Series microcomputers. Refer to the block diagram below.

BLOCK DIAGRAM


P20
P27
8
PORT 2 BUS BUFFER
PORT 2 LATCH (LOW 4) AND EXPANDER PORT I/O
PORT 2 LATCH (HIGH 4)
4
4
HIGH PROGRAM COUNTER (4)
4
3
RESIDENT ROM (8048/49/50 ONLY)
DECODE
8
DB0
DB7
8
EXPANSION TO MORE I/O AND MEMORY
BUS BUFFER
BUS LATCHES AND LOW PC TEMP REG
OSC. FREQ.
÷ BY 3 x 5 x 32
TEST 1
TIMER EVENT COUNTER (8)
LOWER PROGRAM COUNTER (8)
PROGRAM STATUS WORD
PORT 1 BUS BUFFER AND LATCH
8
P10
P17
(8)
ACCUMULATOR (8)
TEMP REG (8)
FLAGS
INSTRUCTION REGISTER AND DECODER
RAM ADDRESS REGISTER
ACCUMULATOR LATCH
ARITHMETIC LOGIC UNIT (8)
DECIMAL ADJUST
CONDITIONAL BRANCH LOGIC
TEST 0
TEST 1
INT
FLAG 0
FLAG 1
TIMER FLAG
CARRY
ACC
ACC BIT TEST
POWER SUPPLY
V_DD RAM SUPPLY
V_CC +5V MAIN SUPPLY
V_SS GND
MULTIPLEXER
DECODER
REGISTER 0
REGISTER 1
REGISTER 2
REGISTER 3
REGISTER 4
REGISTER 5
REGISTER 6
REGISTER 7
8 LEVEL STACK (VARIABLE LENGTH)
OPTIONAL SECOND REGISTER BANK
DATA STORE
RESIDENT RAM ARRAY
CONTROL AND TIMING
T0
INT
RESET
PROG
EA
XTAL1
XTAL2
ALE
PSEN
SS
RD
WR
TIMING OUTPUT
INTERRUPT
INITIALIZE
EXPANDER STROBE
CPU MEMORY SEPARATE
OSCILLATOR XTAL
ADDRESS LATCH ENABLE
PROGRAM MEMORY ENABLE
SINGLE STEP
READ/WRITE STROBES
BD00150S

PROGRAM MEMORY

Resident program memory consists of up to 4K bytes of ROM. The program memory is divided into pages of 256 bytes each. As shown in the memory map, Figure 1, program memory is also divided into two 2048-byte banks, MB0 and MB1. A total of 4096 bytes can be addressed directly. If more memory is required, an I/O port can be used to address locations over 4095.

There are three locations in program memory of special importance. These locations contain the first instruction to be executed upon the occurrence of one of three events.

LOCATION	EVENT
0	Activation then deactivation of the $\overline{\text{RESET}}$ line.
3	Activation of the $\overline{\text{INT}}$ line when the external interrupt is enabled.
7	An overflow of the timer/counter if the T/C interrupt is enabled.

DATA MEMORY

Resident data memory, as shown in Figure 2, consists of up to 256 bytes of RAM. All locations are indirectly addressable by either of two RAM pointer registers at locations 0 and 1. The first eight locations of RAM (0–7) are designated as working registers and are directly addressable by several instructions.

By selecting register bank 1, RAM locations 24–31 become the working registers, replacing those in register bank 0 (0–7).

RAM locations 8–23 are designated as the stack. Two locations (bytes) are used per CALL, allowing nesting of up to eight subroutines.

If additional RAM is required, up to 256 bytes may be added and addressed directly using the MOVX instructions. If more RAM is required an I/O port can be used to select one (256-byte) bank of external memory at a time.

PROGRAM COUNTER AND STACK

The Program Counter (PC) is a 12-bit counter/register that points to the location from which the next instruction is to be fetched. The 8048 and 8049 will automatically address external memory when the boundary of their internal memory is exceeded. All processors access external memory if EA is high.

An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program counter stack. The pair to be used is determined by a 3-bit stack pointer which is part of the Program Status Word (PSW). Data RAM locations 8 through 23 are available as stack registers and are used to store the program counter and 4 bits of PSW. The stack pointer, when initialized to 000, points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to eight times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

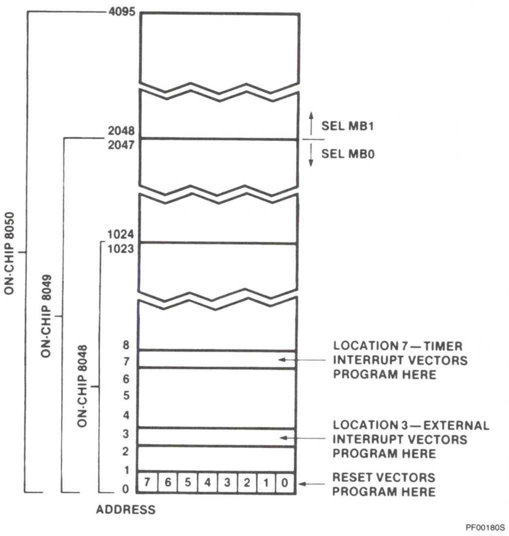

4095
2048
2047
SEL MB1
SEL MB0
1024
1023
ON-CHIP 8050
ON-CHIP 8049
ON-CHIP 8048
8
7
6
5
4
3
2
1
0
7 6 5 4 3 2 1 0
LOCATION 7—TIMER INTERRUPT VECTORS PROGRAM HERE
LOCATION 3—EXTERNAL INTERRUPT VECTORS PROGRAM HERE
RESET VECTORS PROGRAM HERE
ADDRESS
PF00180S

Figure 1. Program Memory Map

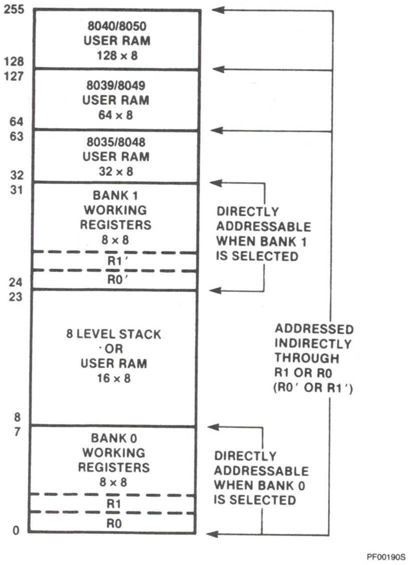
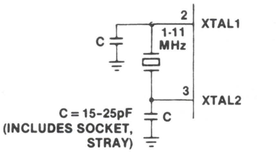

255
8040/8050 USER RAM 128 x 8
128
127
8039/8049 USER RAM 64 x 8
64
63
8035/8048 USER RAM 32 x 8
32
31
BANK 1 WORKING REGISTERS 8 x 8
R1′
R0′
24
23
DIRECTLY ADDRESSABLE WHEN BANK 1 IS SELECTED
8 LEVEL STACK OR USER RAM 16 x 8
ADDRESSED INDIRECTLY THROUGH R1 OR R0 (R0′ OR R1′)
8
7
BANK 0 WORKING REGISTERS 8 x 8
R1
R0
0
DIRECTLY ADDRESSABLE WHEN BANK 0 IS SELECTED
PF00190S

Figure 2. Data Memory Map

OSCILLATOR AND CLOCK

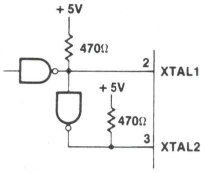
The processor contains its own internal oscillator and clock driver. A crystal, inductor, or external pulse generator may be used to determine the oscillator frequency (see Figure 3). The output of the oscillator is divided by three and can be output on the T0 pin by executing the ENT0 CLK instruction. This CLK signal is divided by 5 to define a machine (instruction) cycle. It is available on Pin 11 as ALE.



SERIES RESONANT, AT CUT CRYSTAL

TC00190S

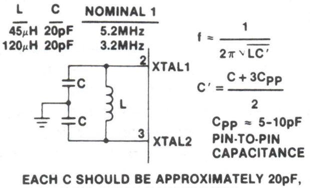
DRIVING FROM EXTERNAL SOURCE



BOTH X1 AND X2 SHOULD BE DRIVEN. RESISTORS TO V_{CC} ARE NEEDED TO ENSURE $V_{IH} = 3.8V$ IF TTL CIRCUITRY IS USED. THE MINIMUM HIGH AND THE MINIMUM LOW TIMES ARE 45%.

TC00200S

LC OSCILLATOR MODE



L	C	NOMINAL 1
45μH	20pF	5.2MHz
120μH	20pF	3.2MHz

$f \approx \frac{1}{2\pi\sqrt{LC'}}$

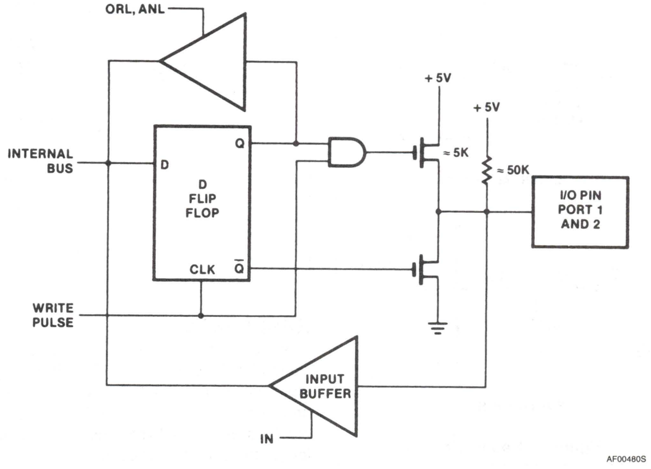
$C' = \frac{C + 3C_{PP}}{2}$

$C_{PP} \approx$ 5-10pF PIN-TO-PIN CAPACITANCE

EACH C SHOULD BE APPROXIMATELY 20pF, INCLUDING STRAY CAPACITANCE.

TC00210S

Figure 3. Crystal Oscillator Mode



AF00480S

Figure 4. "Quasi Bidirectional" Port Structure

TIMER/EVENT COUNTER

An internal counter is available which can count either external events or machine cycles (÷ 32). The machine cycles are divided by 32 before they are input to the 8-bit counter. External events are input directly to the counter. The maximum frequency that can be counted is one third of the frequency of the cycle counter. The minimum positive duty cycle that can be detected is 0.2 t_{CY}. The counter is under program control and can be made to generate an interrupt to the processor when it overflows.

INTERRUPT

An interrupt may be generated by either an external input ($\overline{INT}$, Pin 6) or the overflow of the internal counter, when enabled. In either case, the processor completes execution of the present instruction and then does a CALL to the interrupt service routine. After service, a RETR instruction restores the machine to the state it was prior to the interrupt. The external interrupt has priority over the internal interrupt.

INPUT/OUTPUT

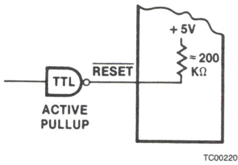
The processor has 27 lines which can be used for input or output functions. These lines are grouped as 3 ports of 8 lines each which serve as either inputs, outputs or bidirectional ports and 3 "test" inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

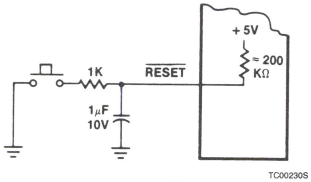
Ports 1 and 2 are each 8 bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching; i.e., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even though outputs are statically latched. Figure 4 shows the circuit configuration. Each line is continuously pulled up to +5V through a resistive device of relatively high impedance (~50K). This pullup is sufficient to provide the source current for a TTL high level yet can be pulled low by a standard TTL gate thus allowing the same pin to be used for both input and output. To provide fast switching times in a "0" to "1" transition a relatively low impedance device (~50KΩ) is switched in momentarily (~500ns) whenever a "1" is written to the line. When a "0" is written to the line, a low impedance (~3000Ω) device overcomes the light pullup and provides TTL current sinking capability.

Since the pulldown transistor is a low impedance device a "1" must first be written to any line which is to be used as an input. Reset initializes all lines to the high impedance "1" state. This structure allows input and output on the same pin and also allows a mix of input lines and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor.



EXTERNAL RESET



POWER ON RESET

Figure 5. Reset Circuits

BUS

BUS is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, BUS can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed.

As a static port, data is written and latched using the OUTL instruction and input using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port, the MOVX instructions are used to read and write to the port. A write to the port generates a pulse on the $\overline{WR}$ output line and output data is valid at the trailing edge of $\overline{WR}$. A read of the port generates a pulse on the $\overline{RD}$ output line and input data must be valid at the trailing edge of $\overline{RD}$. When not being written or read, the BUS lines are in a high impedance state.

Test and INT inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These are T0, T1, and $\overline{INT}$. These pins allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1, and $\overline{INT}$ pins have other possible functions as well.

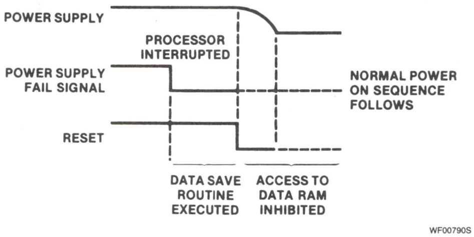

POWER SUPPLY
PROCESSOR INTERRUPTED
POWER SUPPLY FAIL SIGNAL
NORMAL POWER ON SEQUENCE FOLLOWS
RESET
DATA SAVE ROUTINE EXECUTED
ACCESS TO DATA RAM INHIBITED
WF00790S

Figure 6. Power Down Sequence

RESET INPUT

The reset input provides a means for initialization for the processor. This Schmitt-trigger input has an internal pullup resistor which in combination with an external 1μF capacitor provides an internal reset pulse of sufficient length to guarantee all circuitry is reset. If the reset pulse is generated externally, the reset pin must be held at ground (0.5V) for at least 10 milliseconds after the power supply is within tolerance. Only five machine cycles (12.5μs @ 6MHz) are required if power is already on and the oscillator has stabilized. Typical circuitry is shown in Figure 5.

SINGLE STEP

By proper control of the $\overline{SS}$ line, the mircrocomputer can be made to execute one instruction and then pause or wait until the single step switch is activated again.

POWER DOWN MODE

The SCN8049 Series devices permit power to be removed from all but the data RAM array for low power standby operation. In the power down mode the contents of data RAM can be maintained while drawing typically 5% of normal operating power.

V_{CC} serves as the 5V supply pin for the bulk of the circuitry while the V_{DD} pin supplies only the RAM array. In normal operation both pins are at +5V. In standby, V_{CC} is at ground and only V_{DD} is maintained at its specified voltage. Applying $\overline{RESET}$ to the processor through the $\overline{RESET}$ pin inhibits any access to the RAM by the processor and guarantees that RAM cannot be inadvertently altered as power is removed from V_{CC}.

A typical power down sequence occurs as shown in Figure 6.

INSTRUCTION SET

The SCN8049 Series instruction set consists of over 90 one and two byte instructions (see Table 1). Program code efficiency is high because: (1) working registers and program variables are stored in RAM, which require only one byte to address and (2) program memory is divided into pages of 256 bytes each, which means that branch destination addresses require one byte.

The instruction set efficiently manipulates and tests bits in addition to performing logical and arithmetic operations upon and the testing of bytes. A set of move instructions operates indirectly upon either RAM or ROM, which permits efficient access of pointers and data tables. The indirect jump instruction performs a multi (up to 256) way branch upon the content of the accumulator to addresses stored in a lookup table. The "decrement register and jump if not zero" instruction saves a byte every time it is used versus using separate increment and test instructions.

The on-chip counter enables either external events or time to be counted off-line from the main program. The processor can either test the counter (under program control) or cause its overflow to generate an interrupt. These features are highly desirable for real time applications. See Table 2 for instruction timing.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range		
	SCN80xxHC	0 to +70	°C
	SCN80xxHA	−40 to +85	°C
T_{STG}	Storage temperature range	−65 to +150	°C
V_{IN}	Input voltages with respect to V_{SS}[3]	−0.5 to +7	V
P_D	Power dissipation	1.5	W

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$[4, 5, 6]

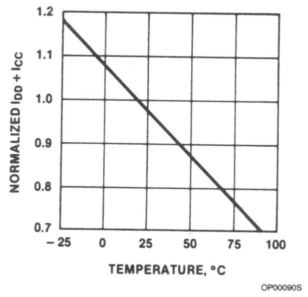
SYMBOL	PARAMETER	TEST CONDITIONS[7]	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low-voltage					
	All except XTAL1, XTAL2		−0.5		0.8	V
V_{IL1}	XTAL1, XTAL2		−0.5		0.6	V
V_{IH}	Input high voltage					
	All except $\overline{\text{RESET}}$, XTAL1, XTAL2		2.0		V_{CC}	V
V_{IH1}	$\overline{\text{RESET}}$, XTAL1, XTAL2		3.8		V_{CC}	V
V_{OL}	Output low-voltage	$I_{OL} = 2.0mA$			0.4	V
V_{OH}	Output high-voltage					
	All except BUS	$I_{OH} = -125\mu A$	2.4			V
	BUS	$I_{OH} = -400\mu A$	2.4			V
I_{LI1}	Port1, Port2, EA, $\overline{\text{SS}}$	$V_{SS} + 0.45 \leq V_{IN} \leq V_{CC}$			−500	μA
I_{L1}	T1, $\overline{\text{Int}}$	$V_{SS} + 0.45 \leq V_{IN} \leq V_{CC}$			±10	μA
I_{LI2}	$\overline{\text{RESET}}$	$V_{SS} + 0.45 \leq V_{IN} \leq V_{CC}$	−10		−300	μA
I_{OL}	Output leakage current					
	BUS, T0 (high inpedance state)	$V_{SS} + 0.45 \leq V_{IN} \leq V_{CC}$			±10	μA
I_{DD}	Standby supply current	$\overline{\text{RESET}} \leq V_{IL}$				
	8035/8048	All inputs = 0V			2.5	mA
	8039/8049	$V_{CC} = 0V$			4.5	mA
	8040/8050				8.5	mA
$I_{DD} + I_{CC}$	Total supply current	$\overline{\text{RESET}} \leq V_{IL}$				
	8035/8048			45	80	mA
	8039/8049			50	95	mA
	8040/8050			60	110	mA
V_{DD}	Standby power supply		2.5			V
$T_A = -40$ to 85°C, Automotive temperature range[8]						
V_{IH}	Input high voltage					
	All except XTAL1 and XTAL2		2.2			V
V_{IH1}	$\overline{\text{RESET}}$, XTAL1, XTAL2		4.0			V
	Input leakage current					
I_{IL1}	Port1, Port2, EA, $\overline{\text{SS}}$	$V_{SS} + .45 \leq V_{IN} \leq V_{CC}$			−750	μA
I_{LI2}	$\overline{\text{RESET}}$	$V_{SS} + .45 \leq V_{IN} \leq V_{CC}$	−5		−300	μA
I_{DD}	Standby supply current					
	8035/8048	$\overline{\text{RESET}} \leq V_{IL}$			3.75	mA
	8039/8049	All inputs = 0V			6.75	mA
	8040/8050	$V_{CC} = 0V$			12.75	mA
$I_{CC} + I_{DD}$	Total supply current					
	8035/8048	$\overline{\text{RESET}} \leq V_{IL}$			90	mA
	8039/8049				105	mA
	8040/8050				120	mA

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$[4, 5, 6]

SYMBOL	PARAMETER	TEST CONDITIONS[7]	11 MHz VERSIONS		6 MHz VERSIONS		UNIT
			Min	Max	Min	Max	
(Refer to Figures 7, 8 and 9)							
t_{LL}	ALE pulse width		150		400		ns
t_{AL}	Address setup to ALE		70		150		ns
t_{LA}	Address hold from ALE		50		80		ns
t_{CC}	Control pulse width ($\overline{\text{PSEN}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$)		300		700		ns
t_{DW}	Data setup before $\overline{\text{WR}}$		250		500		ns
t_{WD}	Data hold after $\overline{\text{WR}}$		40		120		ns
t_{CY}	Cycle time		1.36	3.75	2.5	15.0	µs
t_{DR}	Data hold		0	100	0	200	ns
t_{RD}	$\overline{\text{PSEN}}$, $\overline{\text{RD}}$ to data in			200		500	ns
t_{AW}	Address setup to $\overline{\text{WR}}$		200		230		ns
t_{AD}	Address setup to data in			400		950	ns
t_{AFC}	Address float to $\overline{\text{RD}}$, $\overline{\text{PSEN}}$		−10		0		ns
t_{CA}	Control pulse to ALE		10		10		ns
(Refer to Figure 10)							
t_{CP}	Port control setup before falling edge of PROG		100		110		ns
t_{PC}	Port control hold after falling edge of PROG		60		130		ns
t_{PR}	PROG to time P2 input must be valid			650		810	ns
t_{DP}	Output data setup time		200		250		ns
t_{PD}	Output data hold time		20		65		ns
t_{PF}	Input data hold time		0	150	0	150	ns
t_{PP}	PROG pulse width		700		1200		ns
t_{PL}	Port 2 I/O data setup		250		350		ns
t_{LP}	Port 2 I/O data hold		20		150		ns

NOTES:

1. Stresses above those listed under absolute maximum ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to aviod applying any voltages larger than the rated maximum.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground (V_{SS}). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical ssupply voltages and typical processing parameters.
7. Control outputs: $C_L = 80pF$
   Bus outputs: $C_L = 150pF$
   $t_{CY} = 1.36\mu s$ for 11 MHz versions
   $t_{CY} = 2.5\mu s$ for 6 MHz versions
8. Where no specification is shown, the commercial temperature range specification applies.



NORMALIZED TOTAL SUPPLY CURRENT vs. TEMPERATURE (TYPICAL)

TIMING DIAGRAMS

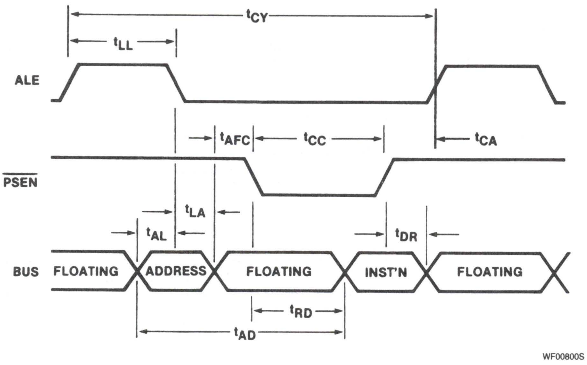

tCY
tLL
ALE
tAFC
tCC
tCA
PSEN
tLA
tAL
tDR
BUS
FLOATING
ADDRESS
FLOATING
INST'N
FLOATING
tRD
tAD
WF00800S

Figure 7. Instruction Fetch From External Program Memory

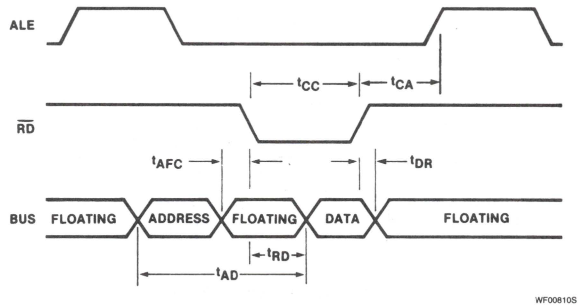

ALE
tCC
tCA
RD
tAFC
tDR
BUS
FLOATING
ADDRESS
FLOATING
DATA
FLOATING
tRD
tAD
WF00810S

Figure 8. Read From External Data Memory

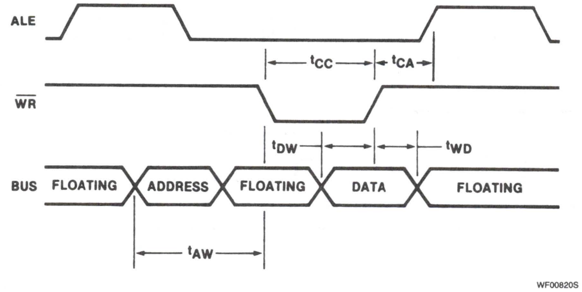

ALE
tCC
tCA
WR
tDW
tWD
BUS
FLOATING
ADDRESS
FLOATING
DATA
FLOATING
tAW
WF00820S

Figure 9. Write to External Data Memory

TIMING DIAGRAMS (Continued)

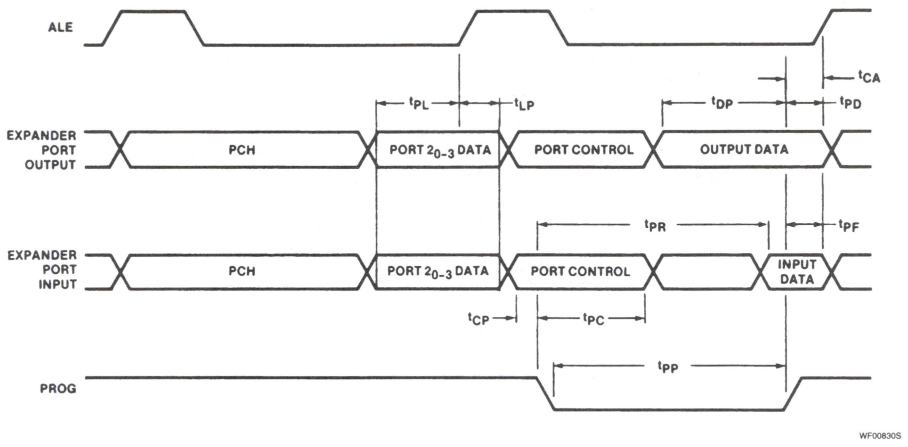
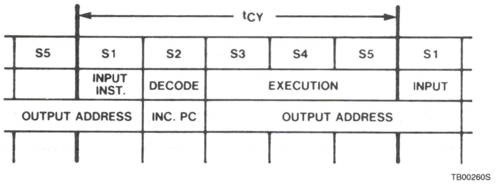

ALE
EXPANDER PORT OUTPUT
PCH
PORT 2_{0-3} DATA
PORT CONTROL
OUTPUT DATA
EXPANDER PORT INPUT
PCH
PORT 2_{0-3} DATA
PORT CONTROL
INPUT DATA
PROG
t_{PL}
t_{LP}
t_{DP}
t_{PD}
t_{CA}
t_{PR}
t_{PF}
t_{CP}
t_{PC}
t_{PP}
WF00830S

Figure 10. Port 2 Timing


t_{CY}
TB00260S

S5	S1	S2	S3	S4	S5	S1
	INPUT INST.	DECODE	EXECUTION			INPUT
OUTPUT ADDRESS		INC. PC	OUTPUT ADDRESS			

Figure 11. Instruction Cycle

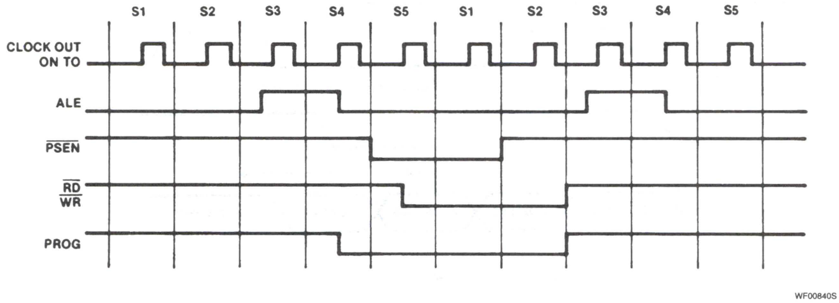

S1
S2
S3
S4
S5
S1
S2
S3
S4
S5
CLOCK OUT ON TO
ALE
$\overline{PSEN}$
$\overline{RD}$
$\overline{WR}$
PROG
WF00840S

Figure 12. Instruction Cycle Timing

Table 1. Instruction Set

MNEMONIC	FUNCTION	DESCRIPTION	INSTRUCTION CODE	CYCLES	BYTES	FLAGS				
			D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0			C	AC	F0	F1	F2
Accumulator										
ADD A, # data	(A) ← (A) + data	Add immediate the specified data to the accumulator.	0 0 0 0 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2	•	•			
ADD A, Rr	(A) ← (A) + (Rr) for r = 0 – 7	Add contents of designated register to the accumulator.	0 1 1 0 1 r r r	1	1	•	•			
ADD A, @ Rr	(A) ← (A) + ((Rr)) for r = 0 – 1	Add indirect the contents the data memory location to the accumulator.	0 1 1 0 0 0 0 r	1	1	•	•			
ADDC A, # data	(A) ← (A) + (C) + data	Add immediate with carry the specified data to the accumulator.	0 0 0 1 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2	•	•			
ADDC A, Rr	(A) ← (A) + (C) + (Rr) for r = 0 – 7	Add with carry the contents of the designated register to the accumulator.	0 1 1 1 1 r r r	1	1	•	•			
ADDC A, @ Rr	(A) ← (A) + (C) + ((Rr)) for r = 0 – 1	Add indirect with carry the contents of data memory location to the accumulator.	0 1 1 1 0 0 0 r	1	1	•	•			
ANL A, # data	(A) ← (A) AND data	Logical AND specified immediate data with accumulator.	0 1 0 1 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
ANL A, Rr	(A) ← (A) AND (Rr) for r = 0 – 7	Logical AND contents of designated register with accumulator.	0 1 0 1 1 r r r	1	1					
ANL A, @ Rr	(A) ← (A) AND ((Rr)) for r = 0 – 1	Logical AND indirect the contents of data memory with accumulator.	0 1 0 1 0 0 0 r	1	1					
CPL A	(A) ← NOT (A)	Complement the contents of the accumulator.	0 0 1 1 0 1 1 1	1	1					
CLR A	(A) ← 0	Clear the contents of the accumulator.	0 0 1 0 0 1 1 1	1	1					
DA A		Decimal adjust the contents of the accumulator.	0 1 0 1 0 1 1 1	1	1	•	•			
DEC A	(A) ← (A) – 1	Decrement the accumulator's contents by 1.	0 0 0 0 0 1 1 1	1	1					
INC A	(A) ← (A) + 1	Increment the accumulator's contents by 1.	0 0 0 1 0 1 1 1	1	1					
ORL A, # data	(A) ← (A) OR data	Logical OR specified immediate data with accumulator.	0 1 0 0 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
ORL A, Rr	(A) ← (A) OR (Rr) for r = 0 – 7	Logical OR contents of designated register with accumulator.	0 1 0 0 1 r r r	1	1					
ORL A, @ Rr	(A) ← (A) OR ((Rr)) for r = 0 – 1	Logical OR indirect the contents of data memory location with accumulator.	0 1 0 0 0 0 0 r	1	1					
RL A	(An + 1) ← (An) (A_0) ← (A_7) for N = 0 ← 6	Rotate accumulator left by 1-bit without carry.	1 1 1 0 0 1 1 1	1	1					
RLC A	(An + 1) ← (An); n = 0 – 6 (A_0) ← (C) (C) ← (A_7)	Rotate accumulator left by 1-bit through carry.	1 1 1 1 0 1 1 1	1	1	•				
RR A	(An) ← (An + 1); n = 0 – 6 (A_7) ← (A_0)	Rotate accumulator right by 1-bit without carry.	0 1 1 1 0 1 1 1	1	1					
RRC A	(An) ← (An + 1); n = 0 – 6 (A_7) ← (C) (C) ← (A_0)	Rotate accumulator right by 1-bit through carry.	0 1 1 0 0 1 1 1	1	1	•				
SWAP A	(A_{4-7}) ← (A_0 – 3)	Swap the 2 4-bit nibbles in the accumulator.	0 1 0 0 0 1 1 1	1	1					
XRL A, # data	(A) ← (A) XOR data	Logical XOR specified immediate data with accumulator.	1 1 0 1 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
XRL A, Rr	(A) ← (A) XOR (Rr) for r = 0 – 7	Logical XOR contents of designated register with accumulator.	1 1 0 1 1 r r r	1	1					
XRL A, @ Rr	(A) ← (A) XOR ((Rr)) for r = 0 – 1	Logical XOR indirect the contents of data memory location with accumulator.	1 1 0 1 0 0 0 r	1	1					

Table 1. Instruction Set (Continued)

MNEMONIC	FUNCTION	DESCRIPTION	INSTRUCTION CODE	CYCLES	BYTES	FLAGS				
			D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0			C	AC	F0	F1	F2
Branch										
DJNZ Rr, addr	(Rr) ← (Rr) − 1; r = 0 − 7 if (Rr) ≠ 0: (PC 0 − 7) ← addr	Decrement the specified register and test contents.	1 1 1 0 1 r r r a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JBb addr	(PC 0 − 7) ← addr if Bb = 1 (PC) ← (PC) + 2 if Bb = 0	Jump to specified address if accumulator bit is set.	b_2 b_1 b_0 1 0 0 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JC addr	(PC 0 − 7) ← addr if C = 1 (PC) ← (PC) + 2 if C = 0	Jump to specified address if carry flag is set.	1 1 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JF0 addr	(PC 0 − 7) ← addr if F0 = 1 (PC) ← (PC) + 2 if F0 = 0	Jump to specified address if flag F0 is set.	1 0 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JF1 addr	(PC 0 − 7) ← addr if F1 = 1 (PC) ← (PC) + 2 if F1 = 0	Jump to specified address if flag F1 is set.	0 1 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JMP addr	(PC 8 − 10) ← addr 8 − 10 (PC 0 − 7) ← addr 0 − 7 (PC 11) ← (DBF)	Direct jump to specified address within the 2K address block.	a_{10} a_9 a_8 0 0 1 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JMPP @ A	(PC 0 − 7) ← ((A))	Jump indirect to specified address within address page.	1 0 1 1 0 0 1 1	2	1					
JNC addr	(PC 0 − 7) ← addr if C = 0 (PC) ← (PC) + 2 if C = 1	Jump to specified address if carry flag is low.	1 1 1 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JNI	(PC 0 − 7) ← addr if $\overline{INT}$ = 0 (PC) ← (PC) + 2 if $\overline{INT}$ = 1	Jump to specified address if $\overline{INT}$ input is low.	1 0 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JNT0 addr	(PC 0 − 7) ← addr if T0 = 0 (PC) ← (PC) + 2 if T0 = 1	Jump to specified address if test 0 is low.	0 0 1 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JNT1 addr	(PC 0 − 7) ← addr if T1 = 0 (PC) ← (PC) + 2 if T1 = 1	Jump to specified address if test 1 is low.	0 1 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JNZ addr	(PC 0 − 7) ← addr if A = 0 (PC) ← (PC) + 2 if A = 0	Jump to specified address if accumulator is non-zero.	1 0 0 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JTF addr	(PC 0 − 7) ← addr if TF = 1 (PC) ← (PC) + 2 if TF = 0	Jump to specified address if timer flag is set to 1.	0 0 0 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JT0 addr	(PC 0 − 7) ← addr if T0 = 1 (PC) ← (PC) + 2 if T0 = 0	Jump to specified address if test 0 is a 1.	0 0 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JT1 addr	(PC 0 − 7) ← addr if T1 = 1 (PC) ← (PC) + 2 if T1 = 0	Jump to specified address if test 1 is a 1.	0 1 0 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
JZ addr	(PC 0 − 7) ← addr if A = 0 (PC) ← (PC) + 2 if A ≠ 0	Jump to specified address if accumulator is 0.	1 1 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
Control										
EN I		Enable the external ($\overline{INT}$) interrupt.	0 0 0 0 0 1 0 1	1	1					
DIS I		Disable the external ($\overline{INT}$) interrupt.	0 0 0 1 0 1 0 1	1	1					
SEL RB0	(BS) ← 0	Select bank 0 (locations 0 − 7) of data memory.	1 1 0 0 0 1 0 1	1	1					•

Table 1. Instruction Set (Continued)

MNEMONIC	FUNCTION	DESCRIPTION	INSTRUCTION CODE	CYCLES	BYTES	FLAGS				
			D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0			C	AC	F0	F1	F2
Control (Cont.)										
SEL RB1	(BS) ← 1	Select bank 1 (locations 24 – 31) of data memory.	1 1 0 1 0 1 0 1	1	1					•
SEL MB0	(DBF) ← 0	Select program memory bank 0, addresses 0 – 2047.	1 1 1 0 0 1 0 1	1	1					
SEL MB1	(DBF) ← 1	Select program memory bank 1, addresses 2048 – 4095	1 1 1 1 0 1 0 1	1	1					
ENT0 CLK		Enable clock output on T0 pin.	0 1 1 1 0 1 0 1	1	1					
Data moves										
MOV A, # data	(A) ← data	Move immediate the specified data into the accumulator.	0 0 1 0 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
MOV A, Rr	(A) ← (Rr); r = 0 – 7	Move the contents of the designated register into the accumulator.	1 1 1 1 1 r r r	1	1					
MOV A, @ Rr	(A) ← ((Rr)); r = 0 – 1	Move indirect the contents of data memory location into the accumulator.	1 1 1 1 0 0 0 r	1	1					
MOV A, PSW	(A) ← (PSW)	Move contents of the program status word into the accumulator.	1 1 0 0 0 1 1 1	1	1					
MOV Rr, # data	(Rr) ← data; r = 0 – 7	Move immediate the specified data into the designated register.	1 0 1 1 1 r r r d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
MOV Rr, A	(Rr) ← (A); r = 0 – 7	Move accumulator contents into the designated register.	1 0 1 0 1 r r r	1	1					
MOV @ Rr, A	((Rr)) ← (A); r = 0 – 1	Move indirect accumulator contents into data memory location.	1 0 1 0 0 0 0 r	1	1					
MOV @ Rr, # data	((Rr)) ← data; r = 0 – 1	Move indirect the specified data into data memory.	1 0 1 1 0 0 0 r d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
MOV PSW, A	(PSW) ← (A)	Move contents of accumulator into the program status word.	1 1 0 1 0 1 1 1	1	1	•	•	•		•
MOVP A, @ A	(A) ← ((A))	Move data in the current page into the accumulator.	1 0 1 0 0 0 1 1	2	1					
MOVP3 A, @ A	(A) ← ((A)) in page 3	Move data in page 3 into the accumulator.	1 1 1 0 0 0 1 1	2	1					
MOVX A, @ Rr	(A) ← ((Rr)); r = 0 – 1	Move indirect the contents of external memory location into the accumulator.	1 0 0 0 0 0 0 r	2	1					
MOVX @ Rr, A	((Rr)) ← (A); r = 0 – 1	Move indirect the contents of the accumulator into external memory.	1 0 0 1 0 0 0 r	2	1					
XCH A, Rr	(A) ↔ (Rr); r = 0 – 7	Exchange the accumulator and designated register's contents.	0 0 1 0 1 r r r	1	1					
XCH A, @ Rr	(A) ↔ ((Rr)); r = 0 – 1	Exchange indirect contents of accumulator and location in data memory.	0 0 1 0 0 0 0 r	1	1					
XCHD A, @ Rr	(A 0 – 3) ↔ (Rr)(0 – 3) r = 0 – 1	Exchange indirect 4-bit contents of accumulator and data memory.	0 0 1 1 0 0 0 r	1	1					
Flags										
CPL C	(C) ← NOT (C)	Complement content of carry bit.	1 0 1 0 0 1 1 1	1	1	•				
CPL F0	(F0) ← NOT (F0)	Complement content of flag F0.	1 0 0 1 0 1 0 1	1	1			•		
CPL F1	(F1) ← NOT (F1)	Complement content of flag F1.	1 0 1 1 0 1 0 1	1	1				•	
CLR C	(C) ← 0	Clear content of carry bit to 0.	1 0 0 1 0 1 1 1	1	1	•				
CLR F0	(F0) ← 0	Clear content of flag 0 to 0.	1 0 0 0 0 1 0 1	1	1			•		
CLR F1	(F1) ← 0	Clear content of flag 1 to 0.	1 0 1 0 0 1 0 1	1	1					•
Input/output										
ANL BUS, # data	(BUS) ← (BUS) AND data	Logical AND immediate specified data with BUS.	1 0 0 1 1 0 0 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
ANL Pp, # data	(Pp) ← (Pp) AND data p = 1 – 2	Logical AND immediate specified data with designated port (1 or 2).	1 0 0 1 1 0 p p d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
ANLD Pp, A	(Pp) ← (Pp) AND (A 0 – 3) p = 4 – 7	Logical AND contents of accumulator with designated port (4 – 7).	1 0 0 1 1 1 p p	2	1					

Table 1. Instruction Set (Continued)

MNEMONIC	FUNCTION	DESCRIPTION	INSTRUCTION CODE	CYCLES	BYTES	FLAGS				
			$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$			C	AC	F0	F1	F2
Input/output (Cont.)										
IN A, Pp	(A) ← (Pp); p = 1 – 2	Input data from designated port (1 – 2) into accumulator.	0 0 0 0 1 0 p p	2	1					
INS A, BUS	(A) ← (BUS)	Input strobed BUS data into accumulator.	0 0 0 0 1 0 0 0	1	2					
MOVD A, Pp	(A 0 – 3) ← (Pp); p = 4 – 7 (A 4 – 7) ← 0	Move contents of designated port (4 – 7) into accumulator.	0 0 0 0 1 1 p p	2	1					
MOVD Pp, A	(Pp) ← A 0 – 3; p = 4 – 7	Move contents of accumulator to designated port (4 – 7).	0 0 1 1 1 1 p p	1	1					
ORLD Pp, A	(Pp) ← (Pp) OR (A 0 – 3) p = 4 – 7	Logical OR contents of accumulator with designated port (4 – 7).	1 0 0 0 1 1 p p	1	1					
ORL BUS, # data	(BUS) ← (BUS) OR data	Logical OR immediate specified data with BUS.	1 0 0 0 1 0 0 0 $d_7\ d_6\ d_5\ d_4\ d_3\ d_2\ d_1\ d_0$	2	2					
ORL Pp, # data	(Pp) ← (Pp) OR data p = 1 – 2	Logical OR immediate specified data with designated port (1 – 2).	1 0 0 0 1 0 p p $d_7\ d_6\ d_5\ d_4\ d_3\ d_2\ d_1\ d_0$	2	2					
OUTL BUS, A	(BUS) ← (A)	Output contents of accumulator onto BUS.	0 0 0 0 0 0 1 0	1	2					
OUTL Pp, A	(Pp) ← (A); p = 1 – 2	Output contents of accumulator to designated port (1 – 2).	0 0 1 1 1 0 p p	1	1					
Registers										
DEC Rr	(Rr) ← (Rr) – 1; r = 0 – 7	Decrement contents of designated register by 1.	1 1 0 0 1 r r r	1	1					
INC Rr	(Rr) ← (Rr) + 1; r = 0 – 7	Increment contents of designated register by 1.	0 0 0 1 1 r r r	1	1					
INC @ Rr	((Rr)) ← ((Rr)) + 1; r = 0 – 1	Increment indirect the contents of data memory location by 1.	0 0 0 1 0 0 0 r	1	1					
Subroutine										
CALL addr	((SP)) ← (PC), (PSW 4 – 7) (SP) ← (SP) + 1 (PC 8 – 10) ← addr 8 – 10 (PC 0 – 7) ← addr 0 – 7 (PC 11) ← DBF	Call designated subroutine.	$a_{10}\ a_9\ a_8$ 1 0 1 0 0 $a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$	2	2					
RET	(SP) ← (SP) – 1 (PC) ← ((SP))	Return from subroutine without restoring program status word.	1 0 0 0 0 0 1 1	2	1					
RETR	(SP) ← (SP) – 1 (PC) ← ((SP)) (PSW 4 – 7) ← ((SP))	Return from subroutine restoring program status word.	1 0 0 1 0 0 1 1	2	1					
Timer/counter										
EN TCNTI		Enable timer/counter interrupt.	0 0 1 0 0 1 0 1	1	1					
DIS TCNTI		Disable timer/counter interrupt.	0 0 1 1 0 1 0 1	1	1					
MOV A, T	(A) ← (T)	Move contents of timer/counter into accumulator.	0 1 0 0 0 0 1 0	1	1					
MOV T, A	(T) ← (A)	Move contents of accumulator into timer/counter.	0 1 1 0 0 0 1 0	1	1					
STOP TCNT		Stop count for event counter or timer.	0 1 1 0 0 1 0 1	1	1					
STRT CNT		Start count for event counter.	0 1 0 0 0 1 0 1	1	1					
STRT T		Start count for timer.	0 1 0 1 0 1 0 1	1	1					
Miscellaneous										
NOP		No operation performed	0 0 0 0 0 0 0 0	1	1					

NOTES:

1. Instruction code designations r and p form the binary representation of the registers and ports involved.
2. The dot under the appropriate flag bit indicates that its content is subject to change by the instruction in which it appears.
3. Numerical subscripts appearing in the FUNCTION column reference the specific bits affected.

SYMBOL DEFINITIONS

SYMBOL	DESCRIPTION
A	The accumulator
AC	The auxiliary carry flag
addr	Program memory address (11 bits)
Bb	Bit designator (b = 0 – 7)
BS	The bank switch
C	Carry flag
CLK	Clock signal
CNT	Event counter
D	Nibble designator (4 bits)
DBF	Program memory bank flip-flop
data	Number or expression (8 bits)
F_0, F_1	Flags 0, 1
I	Interrupt
$\overline{\text{INT}}$	External interrupt
P	"In-Page" operation designator
P_p	Port designator (p = 1, 2 or 4 – 7)
PSW	Program status word
Rr	Register designator (r = 0, 1 or 0 – 7)
SP	Stack pointer
T	Timer
TF	Timer flag
T0, T1	Testable inputs 0, 1
#	Prefix for immediate data
@	Prefix for indirect address
$	Program counter's current value
←	Replaced by
↔	Exchanged with

Table 2. Instruction Timing**

	CYCLE 1					CYCLE 2				
INSTRUCTION	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	Fetch Instruction	Increment Program Counter	—	Increment Timer	—	—	Read Port	* —	—	—
OUTL P,A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Output To Port	—	—	* —	—	—
ANL P, # data	Fetch Instruction	* Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	* Increment Program Counter	Output To Port	—
ORL P, # data	Fetch Instruction	* Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	* Increment Program Counter	Output To Port	—
INS A, BUS	Fetch Instruction	Increment Program Counter	—	Increment Timer	—	—	Read Port	* —	—	—
OUTL BUS, A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Output To Port	—	—	* —	—	—
ANL BUS, # data	Fetch Instruction	* Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	* Increment Program Counter	Output To Port	—
ORL BUS, # data	Fetch Instruction	* Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	* Increment Program Counter	Output To Port	—
MOVX @R,A	Fetch Instruction	Increment Program Counter	Output RAM Address	Increment Timer	Output Data to RAM	—	—	* —	—	—
MOVX A,@R	Fetch Instruction	Increment Program Counter	Output RAM Address	Increment Timer	—	—	Read Data	* —	—	—
MOVD A, P_i	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	—	—	Read P2 Lower	* —	—	—
MOVD P_i, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data TO P2 Lower	—	—	* —	—	—
ANLD P, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data	—	—	* —	—	—
ORLD P, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data	—	—	* —	—	—
J (CONDITIONAL)	Fetch Instruction	* Increment Program Counter	Sample Condition	Increment Timer	—	Fetch Immediate Data	—	* Update Program Counter	—	—
START CNT/STRT T	Fetch Instruction	* Increment Program Counter	—	—	Start Counter					
STOP TCNT	Fetch Instruction	* Increment Program Counter	—	—	Stop Counter					
EN I	Fetch Instruction	* Increment Program Counter	—	Enable Interrupt	—					
DIS I	Fetch Instruction	* Increment Program Counter	—	Disable Interrupt	—					
ENT0 CLK	Fetch Instruction	* Increment Program Counter	—	Enable Clock	—					

NOTES:
*Valid instruction address are output at this time if external program memory is being accessed.
**See figures 11 and 12 for instruction cycle and cycle timing.

Signetics

SCN8031AH/SCN8051AH
Single-Chip 8-Bit Microcontroller

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics SCN8031AH/8051AH is a high-performance microcontroller fabricated using Signetics' highly reliable +5V, depletion-load, N-channel, silicon-gate, N500 MOS process technology. It provides the hardware features, architectural enhancements and instructions that are necessary to make it a powerful and cost-effective controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage.

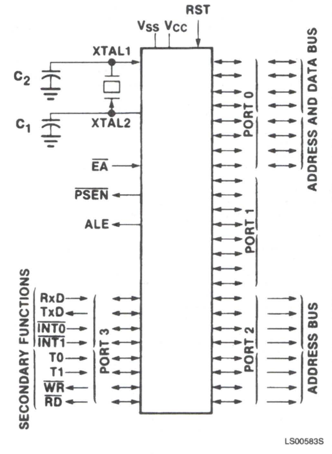
The SCN8051AH contains a 4K × 8 read-only program memory; a 128 × 8 read/write data memory; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. The SCN8031AH is identical, except that it lacks the program memory. For systems that require extra capability, the SCN8051AH can be expanded using standard TTL compatible memories and byte oriented peripheral controllers.

The SCN8051AH microcontroller, like its SCN8048 predecessor, is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12MHz crystal, 58% of the instructions execute in 1μs, 40% in 2μs and multiply and divide require only 4μs.

FEATURES

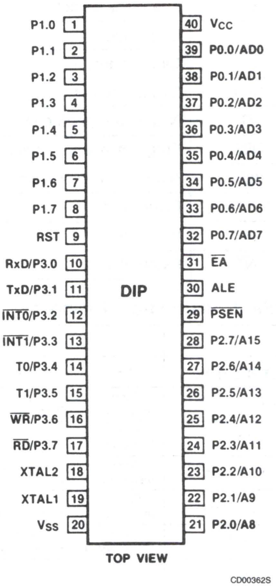
- **Reduced supply current**
- **4K × 8 ROM (SCN8051AH)**
- **128 × 8 RAM**
- **Four 8-bit ports, 32 I/O lines**
- **Two 16-bit timer/event counters**
- **High-performance full-duplex serial channel**
- **External memory expandable to 128K**
- **Boolean processor**
- **Industry standard 8051 architecture:**
  - **Non-paged jumps**
  - **Direct addressing**
  - **Four 8-register banks**
  - **Stack depth up to 128-bytes**
  - **Multiply, divide, subtract, compare**
- **Most instructions execute in 1μs**
- **4μs multiply and divide**

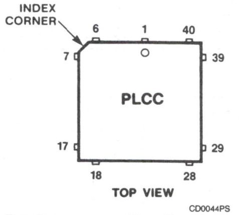
LOGIC SYMBOL


RST
VSS VCC
XTAL1
C2
C1
XTAL2
EA
PSEN
ALE
PORT 0
ADDRESS AND DATA BUS
PORT 1
SECONDARY FUNCTIONS
RxD
TxD
INT0
INT1
T0
T1
WR
RD
PORT 3
PORT 2
ADDRESS BUS
LS00583S

NOTE:
C1, C2 = 30pfd ±10pF for crystals
= 40pfd ±10pF for ceramic resonators

PIN CONFIGURATIONS


P1.0 1
P1.1 2
P1.2 3
P1.3 4
P1.4 5
P1.5 6
P1.6 7
P1.7 8
RST 9
RxD/P3.0 10
TxD/P3.1 11
INT0/P3.2 12
INT1/P3.3 13
T0/P3.4 14
T1/P3.5 15
WR/P3.6 16
RD/P3.7 17
XTAL2 18
XTAL1 19
Vss 20
40 Vcc
39 P0.0/AD0
38 P0.1/AD1
37 P0.2/AD2
36 P0.3/AD3
35 P0.4/AD4
34 P0.5/AD5
33 P0.6/AD6
32 P0.7/AD7
31 EA
30 ALE
29 PSEN
28 P2.7/A15
27 P2.6/A14
26 P2.5/A13
25 P2.4/A12
24 P2.3/A11
23 P2.2/A10
22 P2.1/A9
21 P2.0/A8
DIP
TOP VIEW
CD00362S


INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0	24	A8/P2.0
3	P1.1	25	A9/P2.1
4	P1.2	26	A10/P2.2
5	P1.3	27	A11/P2.3
6	P1.4	28	A12/P2.4
7	P1.5	29	A13/P2.5
8	P1.6	30	A14/P2.6
9	P1.7	31	A15/P2.7
10	RST	32	PSEN
11	RxD/P3.0	33	ALE
12	NC	34	NC
13	TxD/P3.1	35	EA
14	INT0/P3.2	36	AD7/P0.7
15	INT1/P3.3	37	AD6/P0.6
16	T0/P3.4	38	AD5/P0.5
17	T1/P3.5	39	AD4/P0.4
18	WR/P3.6	40	AD3/P0.3
19	RD/P3.7	41	AD2/P0.2
20	XTAL2	42	AD1/P0.1
21	XTAL1	43	AD0/P0.0
22	V_{SS}	44	V_{CC}

Single-Chip 8-Bit Microcontroller SCN8031AH/SCN8051AH

ORDERING INFORMATION

SCN80□□ H □ □ □ □□ (CPxxxx)

ROM/RAM (bytes)
31 = EXT/128
51 = 4K/128

POWER CONSUMPTION
H = Reduced Power AH

OPERATING TEMPERATURE RANGE
A = −40°C to +85°C
C = 0°C to +70°C

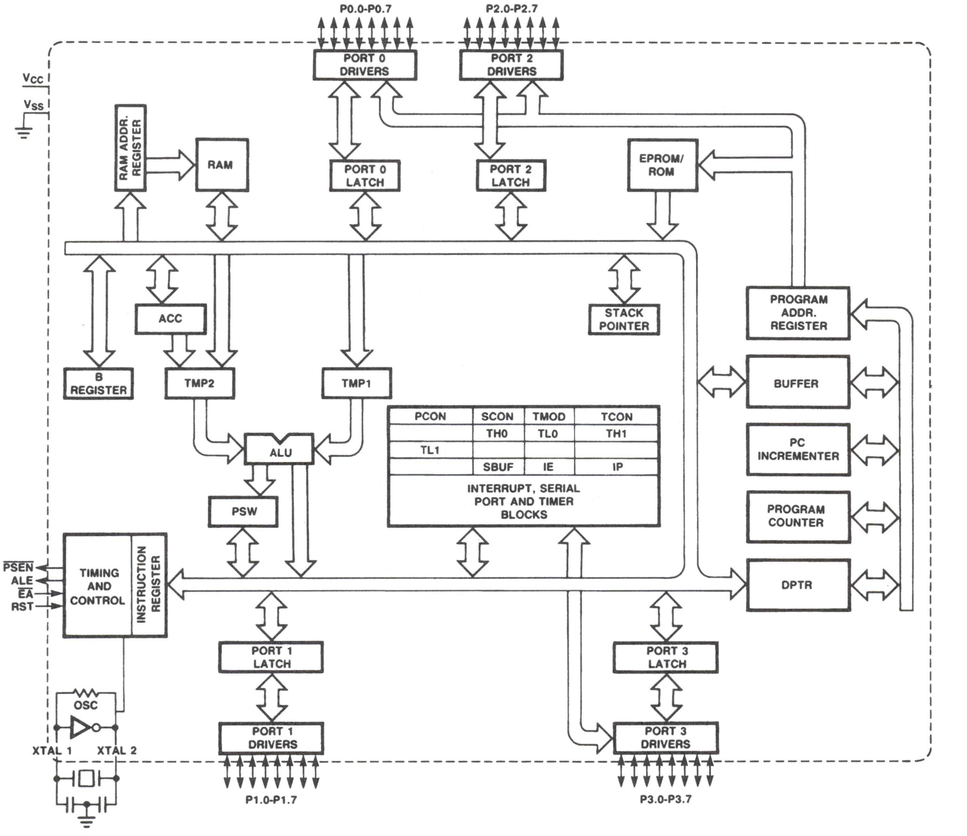
SPEED
C = 12MHz clock

PACKAGE
N = Plastic DIP
I = Ceramic DIP
A = Plastic LCC

40 = PIN DIP
44 = PIN PLCC

CUSTOM ROM PATTERN NUMBER
Applies to masked ROM versions only. Number will be assigned by Signetics. Contact Signetics sales office for ROM pattern submission requirements.

BLOCK DIAGRAM


P0.0-P0.7
P2.0-P2.7
PORT 0 DRIVERS
PORT 2 DRIVERS
VCC
VSS
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
EPROM/ ROM
ACC
STACK POINTER
PROGRAM ADDR. REGISTER
B REGISTER
TMP2
TMP1
BUFFER
PCON
SCON
TMOD
TCON
TH0
TL0
TH1
TL1
SBUF
IE
IP
INTERRUPT, SERIAL PORT AND TIMER BLOCKS
ALU
PC INCREMENTER
PSW
PROGRAM COUNTER
PSEN
ALE
EA
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
PORT 1 LATCH
PORT 3 LATCH
OSC
XTAL 1
XTAL 2
PORT 1 DRIVERS
PORT 3 DRIVERS
P1.0-P1.7
P3.0-P3.7
BD00200S

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	LCC		
V_{SS}	20	22		Circuit ground potential.
V_{CC}	40	44		Supply voltage during normal operation and program verification.
P0.0 – P0.7	39 – 32	43 – 36	I/O	**Port 0:** Port 0 is an 8-bit open-drain, bidirectional I/O port. It is also the multiplexed low-order address and data bus during accesses to external memory. It also outputs instruction bytes during program verification. External pullups are required during program verification. Port 0 can sink (and in bus operation source) eight LS TTL inputs.
P1.0 – P1.7	1 – 8	2 – 9	I/O	**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pullups. It receives the low-order address byte during program verification. Port 1 can sink/source three LS TTL inputs.
P2.0 – P2.7	21 – 28	24 – 31	I/O	**Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pullups. It emits the high-order address byte during accesses to external memory. It also receives the high-order address bits and control signals during program verification. Port 2 can sink/source three LS TTL inputs.
P3.0 – P3.7	10 – 17	11, 13 – 19	I/O	**Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 can sink/source three LS TTL inputs. It also serves the functions of various special features as listed below. The output latch corresponding to a secondary function must be programmed to a logic 1 for that function to operate.
	10	11	I	RXD (P3.0): Serial input port
	11	13	O	TXD (P3.1): Serial output port
	12	14	I	$\overline{\text{INT0}}$ (P3.2): External interrupt 0
	13	15	I	$\overline{\text{INT1}}$ (P3.3): External interrupt 1
	14	16	I	T0 (P3.4): Timer 0 external input
	15	17	I	T1 (3.5): Timer 1 external input
	16	18	O	$\overline{\text{WR}}$ (P3.6): External data memory write strobe
	17	19	O	$\overline{\text{RD}}$ (P3.7): External data memory read strobe
RST	9	10	I	**Reset:** A high-level on this pin for two machine cycles while the oscillator is running resets the device. An external pulldown resistor (≈8.2K) from reset to V_{SS} permits power-on reset using only a capacitor (≈10μF) connected from this pin to V_{CC}.
ALE	30	33	O	**Address Latch Enable:** Output for latching the low byte of the address during accesses to external memory. ALE is activated at a constant rate of 1/6 the oscillator frequency except during an external data memory access, at which time one ALE pulse is skipped. ALE can sink/source eight LS TTL inputs.
$\overline{\text{PSEN}}$	29	32	O	**Program Store Enable:** Output is the read strobe to external program memory. $\overline{\text{PSEN}}$ is activated twice each machine cycle during fetches from external program memory. (However, when executing out of external program memory, two activations of $\overline{\text{PSEN}}$ are skipped during each access to external data memory.) $\overline{\text{PSEN}}$ is not activated during fetches from internal program memory. $\overline{\text{PSEN}}$ can sink/source eight LS TTL inputs.
$\overline{\text{EA}}$	31	35	I	**Instruction Execution Control:** When $\overline{\text{EA}}$ is held high, the CPU executes out of internal program memory (unless the program counter exceeds 0FFFH). When $\overline{\text{EA}}$ is held low, the CPU executes only out of external program memory. $\overline{\text{EA}}$ must not be floated.
XTAL1	19	21	I	**Crystal 1:** Input to the inverting amplifier that forms the oscillator. This pin should be connected to V_{SS} when an external oscillator is used.
XTAL2	18	20	O	**Crystal 2:** Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. XTAL2 receives the oscillator signal when an external oscillator is used.

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2] range SCN80xxHCC SCN80xxHAC	 0 to 70 −40 to +85	 °C °C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +7.0	V
P_D	Power dissipation	1.0	W

DC ELECTRICAL CHARACTERISTICS T_A=0°C to +70°C, V_{CC} = 4.5 to 5.5V, V_{SS} = 0V[4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{IL}	Input low voltage		−0.5	0.8	V
V_{IH}	Input high voltage (except RST and XTAL2)		2	V_{CC} + 0.5	V
V_{IH1}	Input high voltage to RST for reset, XTAL2	XTAL1 to V_{SS}	2.5	V_{CC} + 0.5	V
V_{OL}	Output low voltage ports 1, 2, 3[6]	I_{OL} = 1.6mA		0.45	V
V_{OL1}	Output low voltage port 0, ALE, $\overline{PSEN}$[6]	I_{OL} = 3.2mA		0.45	V
V_{OH}	Output high voltage ports 1, 2, 3	I_{OH} = −80μA	2.4		V
V_{OH1}	Output high voltage port 0, ALE $\overline{PSEN}$	I_{OH} = −400μA	2.4		V
I_{IL}	Logical 0 input current ports 1, 2, 3	V_{IN} = 0.45V		−500	μA
I_{IH1}	Input high current to RST for reset	$V_{IN} < V_{CC} - 1.5V$		500	μA
I_{LI}	Input leakage current to port 0, $\overline{EA}$	$0.45 < V_{IN} < V_{CC}$		±10	μA
I_{IL2}	Logical 0 input current for XTAL2	XTAL1 = V_{SS}, V_{IN} = 0.45V		−3.2	mA
I_{CC}	Power supply current	All outputs disconnected and $\overline{EA}$ = V_{CC}		125	mA
C_{IO}	Capacitance of I/O buffer	f_C = 1MHz, T_A = 25°C		10	pF

T_A = −40°C to +85°C – Extended temperature range – SCN8051HAC only

V_{IH}	Input high voltage (except RST and XTAL2)		2.2	V_{CC}+0.5	V
V_{IH1}	Input high voltage to RST for reset, XTAL2	XTAL1 to V_{SS}	2.7		V
I_{CC}	Power supply current	All outputs disconnected and $\overline{EA}$ = V_{CC}		175	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. For testing, all input signals swing between 0.45V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and at output voltages of 0.8V and 2V as appropriate.
6. V_{OL} is degraded when the SCN8031AH/SCN8051AH rapidly disch arges external capacitance. This AC noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close to the SCN8031AH/SCN8051AH as possible.

Datum	Emitting Ports	Degraded I/O Lines	V_{OL}(Peak Max)
Address	P2, P0	P1, P3	0.8V
Write data	P0	P1, P3, ALE	0.8V

7. C_L = 100pF for port 0, ALE and PSEN outputs; C_L = 80pF for all other ports.

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 10\%$, $V_{SS} = 0V$[4, 5, 7]

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK $1/t_{CLCL}$ = 3.5MHz to 12MHz		UNIT
		Min	Max	Min	Max	
Program memory char (Figure 1)						
t_{LHLL}	ALE pulse width	127		$2t_{CLCL} -40$		ns
t_{AVLL}	Address setup to ALE	43		$t_{CLCL} -40$		ns
t_{LLAX}	Address hold after ALE	48		$t_{CLCL} -35$		ns
t_{LLIV}	ALE to valid instr in		223		$4t_{CLCL} -100$	ns
t_{LLPL}	ALE to $\overline{\text{PSEN}}$	58		$t_{CLCL} -25$		ns
t_{PLPH}	$\overline{\text{PSEN}}$ pulse width	215		$3t_{CLCL} -35$		ns
t_{PLIV}	$\overline{\text{PSEN}}$ to valid instr in		125		$3t_{CLCL} -125$	ns
t_{PXIX}	Input instr hold after $\overline{\text{PSEN}}$	0		0		ns
t_{PXIZ}	Input instr float after $\overline{\text{PSEN}}$		63		$t_{CLCL} -20$	ns
t_{PXAV}	Address valid after $\overline{\text{PSEN}}$	75		$t_{CLCL} -8$		ns
t_{AVIV}	Address to valid instr in		302		$5t_{CLCL} -115$	ns
t_{PLAZ}	$\overline{\text{PSEN}}$ to address float		20		20	ns
External data memory char (Figures 2 and 3)						
t_{RLRH}	$\overline{\text{RD}}$ pulse width	400		$6t_{CLCL} -100$		ns
t_{WLWH}	$\overline{\text{WR}}$ pulse width	400		$6t_{CLCL} -100$		ns
t_{LLAX}	Address hold after ALE	48		$t_{CLCL} -35$		ns
t_{RLDV}	$\overline{\text{RD}}$ to valid data in		252		$5t_{CLCL} -165$	ns
t_{RHDX}	Data hold after $\overline{\text{RD}}$	0		0		ns
t_{RHDZ}	Data float after $\overline{\text{RD}}$		97		$2t_{CLCL} -70$	ns
t_{LLDV}	ALE to valid data in		517		$8t_{CLCL} -150$	ns
t_{AVDV}	Address to valid data in		585		$9t_{CLCL} -165$	ns
t_{LLWL}	ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	200	300	$3t_{CLCL} -50$	$3t_{CLCL} +50$	ns
t_{AVWL}	Address to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	203		$4t_{CLCL} -130$		ns
t_{WHLH}	$\overline{\text{WR}}$ or $\overline{\text{RD}}$ high to ALE high	43	123	$t_{CLCL} -40$	$t_{CLCL} +40$	ns
t_{DVWX}	Data valid to $\overline{\text{WR}}$ transition	23		$t_{CLCL} -60$		ns
t_{QVWH}	Data setup before $\overline{\text{WR}}$	433		$7t_{CLCL} -150$		ns
t_{WHQX}	Data hold after $\overline{\text{WR}}$	33		$t_{CLCL} -50$		ns
t_{RLAZ}	Address float after $\overline{\text{RD}}$		20		20	ns
External clock (Figure 4)						
t_{CLCL}	Oscillator period			83.3	286	ns
t_{CHCX}	High time			20		ns
t_{CLCX}	Low time			20		ns
t_{CLCH}	Rise time				20	ns
t_{CHCL}	Fall time				20	ns

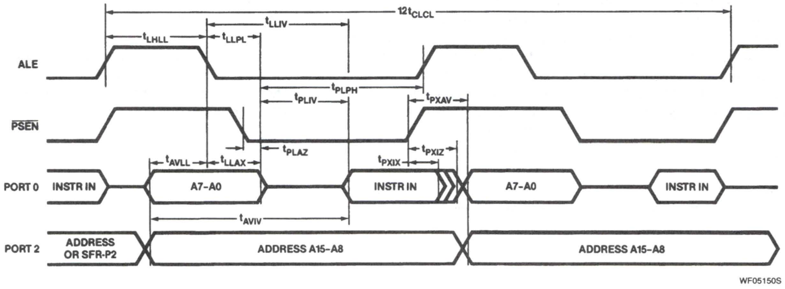


Figure 1. Program Memory Read Cycle

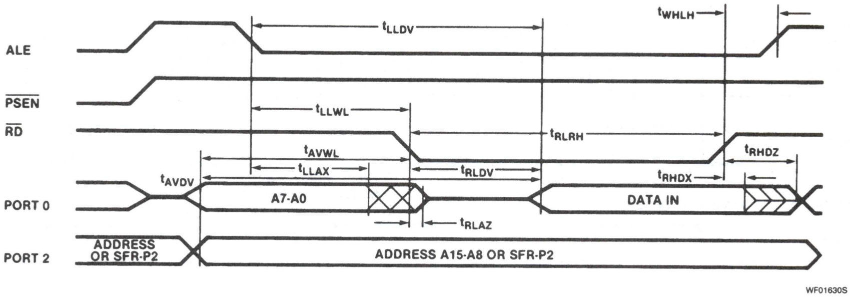

ALE
PSEN
RD
PORT 0
PORT 2
tLLDV
tWHLH
tLLWL
tRLRH
tAVWL
tLLAX
tRLDV
tRHDZ
tAVDV
tRHDX
A7-A0
DATA IN
tRLAZ
ADDRESS OR SFR-P2
ADDRESS A15-A8 OR SFR-P2
WF01630S

Figure 2. Data Memory Read Cycle

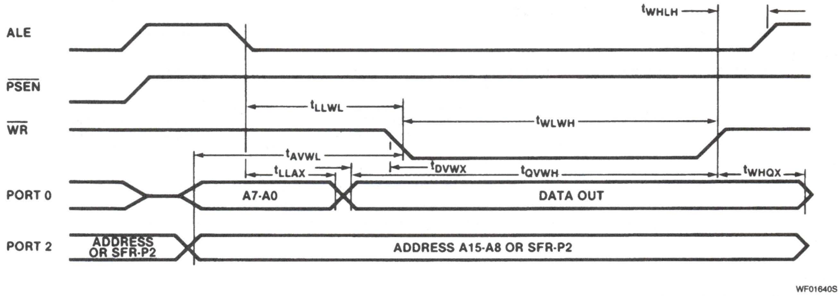

ALE
PSEN
WR
PORT 0
PORT 2
tWHLH
tLLWL
tWLWH
tAVWL
tLLAX
tDVWX
tQVWH
tWHQX
A7-A0
DATA OUT
ADDRESS OR SFR-P2
ADDRESS A15-A8 OR SFR-P2
WF01640S

Figure 3. Data Memory Write Cycle

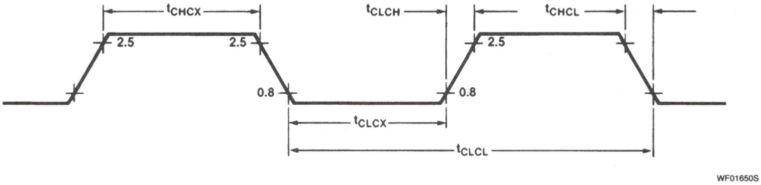
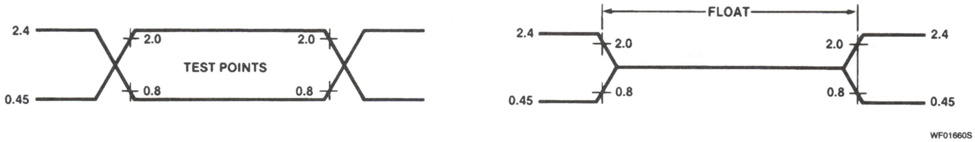
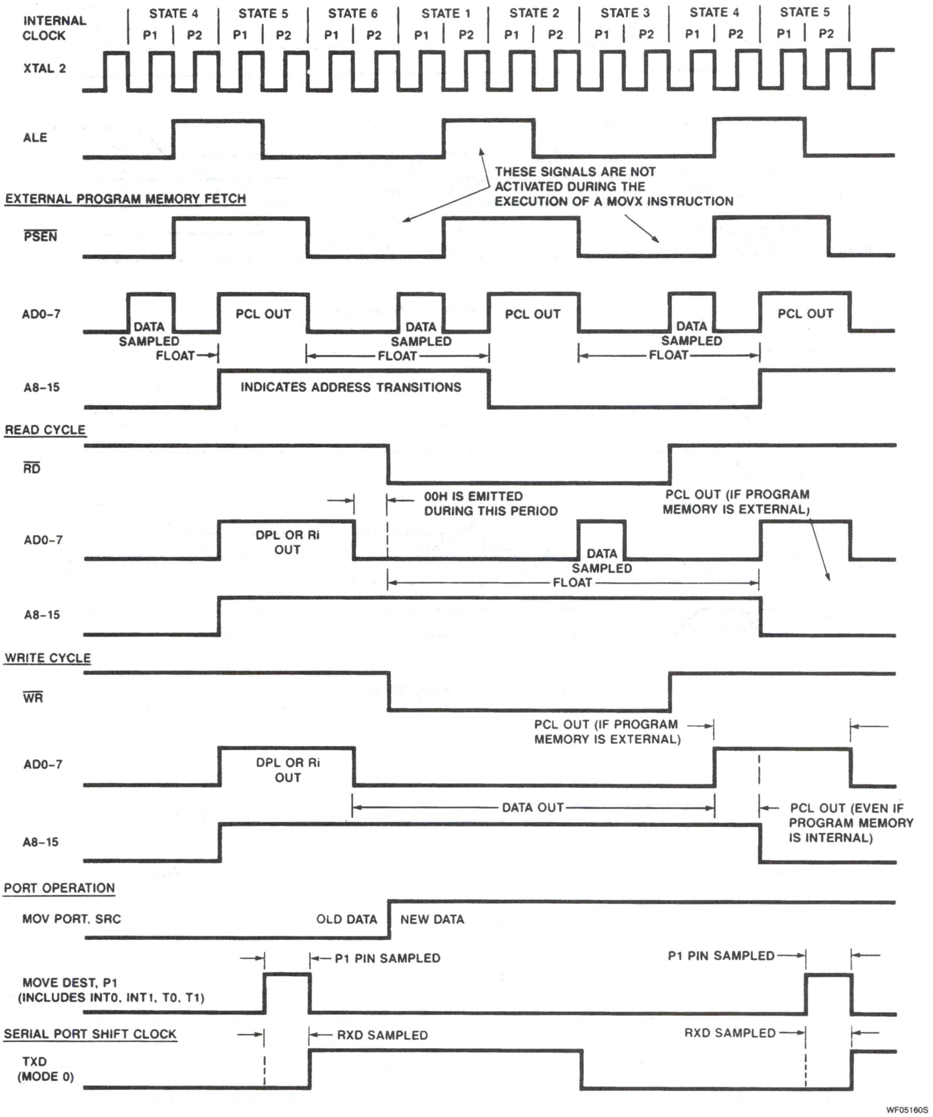

tCHCX
tCLCH
tCHCL
2.5
2.5
2.5
0.8
0.8
tCLCX
tCLCL
WF01650S

Figure 4. External Clock Drive XTAL2


2.4
0.45
2.0
2.0
TEST POINTS
0.8
0.8
FLOAT
2.4
2.0
2.0
2.4
0.45
0.8
0.8
0.45
WF01660S

NOTES:
1. AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0".
2. Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".
3. For timing purposes, the float state is defined as the point at which a P0 pin sinks 3.2mA or sources 400μA at the voltage test levels.

Figure 5. AC Testing Input/Output, Float Waveforms


INTERNAL CLOCK
STATE 4
STATE 5
STATE 6
STATE 1
STATE 2
STATE 3
STATE 4
STATE 5
P1
P2
XTAL 2
ALE
THESE SIGNALS ARE NOT ACTIVATED DURING THE EXECUTION OF A MOVX INSTRUCTION
EXTERNAL PROGRAM MEMORY FETCH
PSEN
AD0-7
DATA SAMPLED
PCL OUT
FLOAT
A8-15
INDICATES ADDRESS TRANSITIONS
READ CYCLE
RD
00H IS EMITTED DURING THIS PERIOD
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL)
AD0-7
DPL OR Ri OUT
DATA SAMPLED
FLOAT
A8-15
WRITE CYCLE
WR
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL)
AD0-7
DPL OR Ri OUT
DATA OUT
PCL OUT (EVEN IF PROGRAM MEMORY IS INTERNAL)
A8-15
PORT OPERATION
MOV PORT, SRC
OLD DATA
NEW DATA
P1 PIN SAMPLED
MOVE DEST, P1 (INCLUDES INT0, INT1, T0, T1)
SERIAL PORT SHIFT CLOCK
RXD SAMPLED
TXD (MODE 0)
WF05160S

NOTE:
All internal timing is referenced to the internal time states shown at the top of the page. This waveform represents the signal on the X2 input of the oscillator. This diagram represents when these signals are actually clocked within the chip. However, the time it takes a signal to propagate to the pins is in the range of 50-150ns. Propagation delays are dependent on many variables, such as temperature and pin loading. Even the different signals vary. Typically though, RD and WR have propagation delays of approximately 50ns and the other timing signals approximately 85ns. At room temperature, fully loaded, these differences in propagation delays between signals have been integrated into the timing specs.

Figure 6. Timing Waveforms

Table 1. Instruction Set

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Arithmetic Operations				
ADD	A,Rn	Add register to accumulator	1	1
ADD	A,direct	Add direct byte to accumulator	2	1
ADD	A,@Ri	Add indirect RAM to accumulator	1	1
ADD	A,R#data	Add immediate data to accumulator	2	1
ADDC	A,Rn	Add register to accumulator with carry	1	1
ADDC	A,direct	Add direct byte to A with carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1
ADDC	A,#data	Add immediate data to A with carry flag	2	1
SUBB	A,Rn	Subtract register from A with borrow	1	1
SUBB	A,direct	Subtract direct byte from A with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A w/borrow	1	1
SUBB	A,#data	Subtract immed. data from A w/borrow	2	1
INC	A	Increment accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement Indirect RAM	1	1
INC	DPTR	Increment data pointer	1	2
MUL	AB	Mulitiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal adjust accumulator	1	1
Logical Operations				
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate A left through the carry flag	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate A right through carry flag	1	1
SWAP	A	Swap nibbles within the accumulator	1	1
Data Transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct, #data	Move immediate data to direct byte	3	2

Table 1. Instruction Set (Continued)

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Data Transfer (Continued)				
MOV	@Ri,A	Move accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to A	1	2
MOVC	A,@A + PC	Move code byte relative to PC to A	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to external RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order digit ind. RAM w/A	1	1
Boolean Variable Manipulation				
CLR	C	Clear carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry flag	2	2
ANL	C,bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry flag	2	2
ORL	C,bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry flag	2	1
MOV	bit,C	Move carry flag to direct bit	2	2
Program and Machine Control				
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long jump	3	2
SJMP	rel	Short jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if accumulator is zero	2	2
JNZ	rel	Jump if accumulator is not zero	2	2
JC	rel	Jump if carry flag is set	2	2
JNC	rel	Jump if no carry flag	2	2
JB	bit,rel	Jump if direct bit set	3	2
JNB	bit,rel	Jump if direct bit not set	3	2
JBC	bit,rel	Jump if direct bit is set & clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & jump if not equal	3	2
CJNE	A,#data, rel	Comp. immed. to A & jump if not equal	3	2
CJNE	Rn,#data, rel	Comp. immed. to reg. & jump if not equal	3	2
CJNE	@Ri,#data, rel	Comp. immed. to ind. & jump if not equal	3	2
DJNZ	Rn,rel	Decrement register & jump if not zero	2	2
DJNZ	direct,rel	Decrement direct & jump if not zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn	–Working register R0-R7
direct	–128 internal RAM locations, any I/O port, control or status register
@Ri	–Indirect Internal RAM location addressed by register R0 or R1
#data	–8-bit constant included in instruction
#data16	–16-bit constant included as bytes 2 & 3 of instruction
bit	–128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr16	–Destination address for LCALL & LJMP may be anywhere within the 64-kilobyte program memory address space.
addr11	–Destination address for ACALL & AJMP will be within the same 2-kilobyte page of program memory as the first byte of the following instruction.
rel	–SJMP and all conditional jumps include an 8-bit offset byte. Range is +127 – 128 bytes relative to first byte of the following instruction.

Table 2. Instruction Opcodes in Hexadecimal Order

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,#data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr,A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data

Table 2. Instruction Opcodes in Hexadecimal Order (Continued)

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A + PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A + DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Signetics

SCN8032AH, SCN8052AH
Single-chip 8-Bit Microcontroller

Preliminary Specification

Microprocessor Products

DESCRIPTION

The Signetics SCN8032AH/SCN8052AH 8-bit Microcontroller is a high-performance single-chip computer, fabricated with Signetics' highly-reliable +5V, depletion-load, N-Channel, silicon-gate MOS technology. It provides the hardware features, architectural enhancements and instructions that are necessary to make it a powerful and cost effective controller for applications.

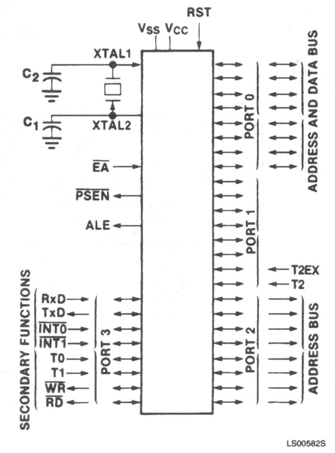
The SCN8032AH contains 256 bytes of read/write data memory; 32 I/O lines configured as four 8-bit ports; three 16-bit timer/counters; a six-source, two-priority level nested interrupt structure; a programmable serial I/O port; and an on-chip oscillator with clock circuitry. The SCN8052AH has all of these features plus 8K bytes of nonvolatile read-only program memory. Both microcontrollers have memory expansion capabilities of up to 64K bytes of data storage and 64K bytes of program memory that may be realized with standard TTL compatible memories.

Because of its' extensive BCD/binary arithmetic and bit-handling facilities, the SCN8032AH/SCN8052AH microcontroller is efficient at both computational and control-oriented tasks. Efficient use of program memory is also achieved by using the familiar compact instruction set of the 8031/8051. Forty-four percent of the instructions are one-byte, 41% two-byte and 15% three-byte. The majority of the instructions execute in just 1.0µs at 12MHz operation. The longest instructions, multiply and divide, require only 4µs at 12MHz.

FEATURES

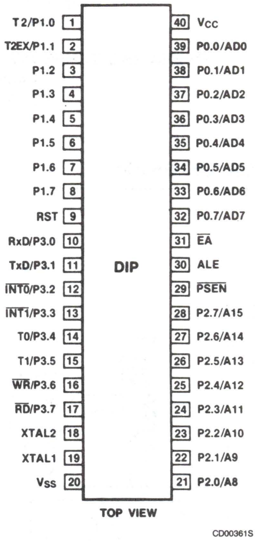
- **SCN8032AH – control-oriented CPU with RAM and I/O**
- **SCN8052AH – an SCN8032AH with factory mask-programmable ROM**
- **8K × 8 ROM (SCN8052AH only)**
- **256 × 8 RAM**
- **32 I/O lines (four 8-bit ports)**
- **Three 16-bit timer/counters**
- **Programmable full-duplex serial channel**
  - **Variable transmit/receive baud rate capability**
- **Timer 2 capture capability**
- **External memory addressing**
  - **64K ROM and 64K RAM**
- **Boolean processor**
- **128 user bit-addressable locations**
- **Upward compatible with SCN8031AH/SCN8051AH**

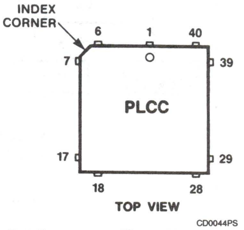
LOGIC SYMBOL


RST
VSS VCC
XTAL1
C2
C1
XTAL2
EA
PSEN
ALE
PORT 0
ADDRESS AND DATA BUS
PORT 1
T2EX
T2
SECONDARY FUNCTIONS
RxD
TxD
INT0
INT1
T0
T1
WR
RD
PORT 3
PORT 2
ADDRESS BUS
LS00582S

NOTES:
C1, C2 = 30 pFd. ±10pFd for crystals
= 40 pFd ±10pFd for ceramic resonators

PIN CONFIGURATIONS


T2/P1.0 1
T2EX/P1.1 2
P1.2 3
P1.3 4
P1.4 5
P1.5 6
P1.6 7
P1.7 8
RST 9
RxD/P3.0 10
TxD/P3.1 11
INT0/P3.2 12
INT1/P3.3 13
T0/P3.4 14
T1/P3.5 15
WR/P3.6 16
RD/P3.7 17
XTAL2 18
XTAL1 19
Vss 20
40 Vcc
39 P0.0/AD0
38 P0.1/AD1
37 P0.2/AD2
36 P0.3/AD3
35 P0.4/AD4
34 P0.5/AD5
33 P0.6/AD6
32 P0.7/AD7
31 EA
30 ALE
29 PSEN
28 P2.7/A15
27 P2.6/A14
26 P2.5/A13
25 P2.4/A12
24 P2.3/A11
23 P2.2/A10
22 P2.1/A9
21 P2.0/A8
DIP
TOP VIEW
CD00361S


INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

Pin	Function	Pin	Function
1	NC	23	NC
2	T2/P1.0	24	A8/P2.0
3	T2EX/P1.1	25	A9/P2.1
4	P1.2	26	A10/P2.2
5	P1.3	27	A11/P2.3
6	P1.4	28	A12/P2.4
7	P1.5	29	A13/P2.5
8	P1.6	30	A14/P2.6
9	P1.7	31	A15/P2.7
10	RST	32	PSEN
11	RXD/P3.0	33	ALE
12	NC	34	NC
13	TXD/P3.1	35	EA
14	INT0/P3.2	36	AD7/P0.7
15	INT1/P3.3	37	AD6/P0.6
16	T0/P3.4	38	AD5/P0.5
17	T1/P3.5	39	AD4/P0.4
18	WR/P3.6	40	AD3/P0.3
19	RD/P3.7	41	AD2/P0.2
20	XTAL2	42	AD1/P0.1
21	XTAL1	43	AD0/P0.0
22	V_{SS}	44	V_{CC}

Single-chip 8-Bit Microcontroller

SCN8032AH, SCN8052AH

ORDERING INFORMATION

SCN80□□ H □ □ □ □□ (CPxxxx)

ROM/RAM (bytes)
32 = EXT/256
52 = 8K/256

POWER CONSUMPTION
H = Reduced Power AH

OPERATING TEMPERATURE RANGE
A = −40°C to +85°C
C = 0°C to +70°C

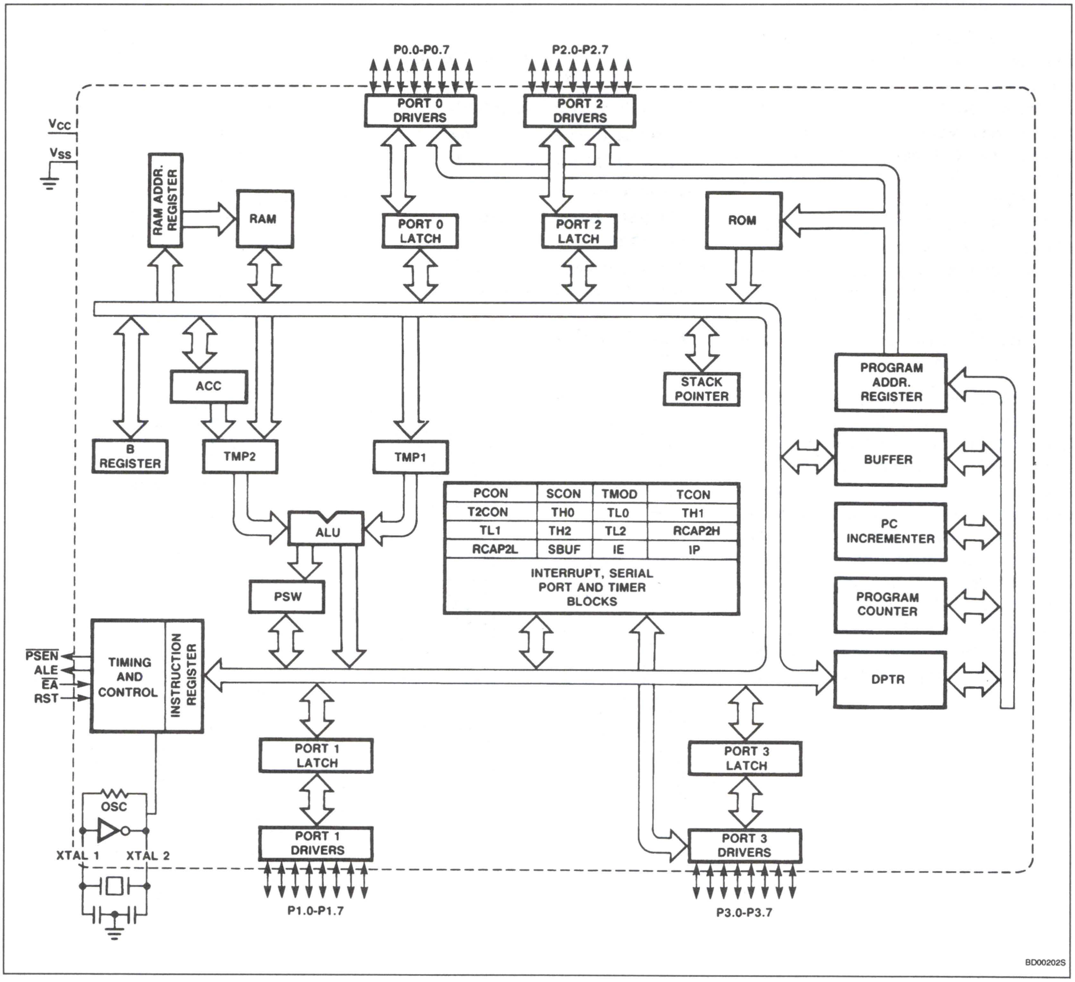
SPEED
C = 12MHz clock

CUSTOM ROM PATTERN NUMBER
Applies to masked ROM versions only. Number will be assigned by Signetics. Contact Signetics sales office for ROM pattern submission requirements.

40 = PIN DIP
44 = PIN PLCC

PACKAGE
N = Plastic DIP
I = Ceramic DIP
A = Plastic LCC

BLOCK DIAGRAM


P0.0-P0.7
P2.0-P2.7
PORT 0 DRIVERS
PORT 2 DRIVERS
VCC
VSS
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
ROM
STACK POINTER
PROGRAM ADDR. REGISTER
ACC
B REGISTER
TMP2
TMP1
BUFFER
PCON
SCON
TMOD
TCON
T2CON
TH0
TL0
TH1
TL1
TH2
TL2
RCAP2H
RCAP2L
SBUF
IE
IP
INTERRUPT, SERIAL PORT AND TIMER BLOCKS
ALU
PC INCREMENTER
PSW
PROGRAM COUNTER
PSEN
ALE
EA
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
PORT 1 LATCH
PORT 3 LATCH
OSC
XTAL 1
XTAL 2
PORT 1 DRIVERS
PORT 3 DRIVERS
P1.0-P1.7
P3.0-P3.7
BD00202S

Single-chip 8-Bit Microcontroller SCN8032AH, SCN8052AH

PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	LCC		
V_{SS}	20	22		Circuit ground potential.
V_{CC}	40	44		+5V supply voltage during operation and program verification.
P0.0 – P0.7	39 – 32	43 – 36	I/O	**Port 0:** Port 0 is an 8-bit open-drain, bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink (and in bus operations can source) eight LS TTL loads.
P1.0 – P1.7	1 – 8	2 – 9	I/O	**Port 1:** Port 1 is an 8-bit bidirectional I/O port. Pins P1.0 and P1.1 also correspond to the special functions T2, timer 2 counter trigger input, and T2EX, external input to timer 2. The output latch on these two special function pins must be programmed to a one for that function to operate. Port 1 is also used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads.
	1	2	I	**T2 (P1.0):** Timer counter 2 trigger input.
	2	3	I	**T2EX (P1.1):** Timer counter 2 external count input.
P2.0 – P2.7	21 – 28	24 – 31	I/O	**Port 2:** Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.
P3.0 – P3.7	10 – 17	11, 13 – 19	I/O	**Port 3:** Port 3 is an 8-bit bidirectional I/O port. The output latch corresponding to a secondary function must be programmed to a one for that function to operate. Port 3 can sink/source four LS TTL loads. Each of the P3 pins also correspond to the special functions listed below:
	10	11	I	**RXD (P3.0):** Serial input port
	11	13	O	**TXD (P3.1):** Serial output port
	12	14	I	**INT0 (P3.2):** External interrupt
	13	15	I	**INT1 (P3.4):** External interrupt
	14	16	I	**T0 (P3.4):** Timer 0 external input
	15	17	I	**T1 (P3.5):** Timer 1 external input
	16	18	O	**WR (P3.6):** External data memory write strobe
	17	19	I	**RD (P3.7):** External data memory read strobe
RST	9	10	I	**Reset:** A high on this pin for two machine cycles while the oscillator is running resets the device. A small external pulldown resistor (≈8.2KΩ) from RST to V_{SS} permits power-on reset when a capacitor (≈10μF) is also connected from this pin to V_{CC}.
ALE	30	33	O	**Address Latch Enable:** Output for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{PSEN}$	29	32	O	**Program Store Enable:** Output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.
$\overline{EA}$	31	35	I	**Instruction Execution Control:** When held at a high TTL level, the SCN8052AH executes instructions from the internal ROM when the PC is less than 8192. When held at a TTL low level, the SCN8032AH/SCN8052AH fetches all instructions from external program memory.
XTAL1	19	21	I	**Crystal 1:** Input to the inverting amplifier that forms the oscillator.
XTAL2	18	20	O	**Crystal 2:** Output of the inverting amplifier that forms the oscillator and input to the internal clock generator. Receives the external oscillator signal when an external oscillator is used.

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature range		
	SCN80xxHCC	0 to 70	°C
	SCN80xxHAC	−40 to +85	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	All voltages with respect to ground[3]	−0.5 to +7.0	V
P_D	Power dissipation	2.0	W

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 4.5$ to $5.5V$, $V_{SS} = 0V$[4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{IL}	Input low voltage		−0.5	0.8	V
V_{IH}	Input high voltage (except RST and XTAL2)		2	$V_{CC} + 0.5$	V
V_{IH1}	Input high voltage to RST for reset, XTAL2	XTAL1 to V_{SS}	2.5	$V_{CC} + 0.5$	V
V_{OL}	Output low voltage ports 1, 2, 3[6]	$I_{OL} = 1.6mA$		0.45	V
V_{OL1}	Output low voltage port 0, ALE, $\overline{\text{PSEN}}$[7]	$I_{OL} = 3.2mA$		0.45	V
V_{OH}	Output high voltage ports 1, 2, 3	$I_{OH} = -80\mu A$	2.4		V
V_{OH1}	Output high voltage port 0, ALE, $\overline{\text{PSEN}}$	$I_{OH} = -400\mu A$	2.4		V
I_{IL}	Logical 0 input current ports 1, 2, 3	$V_{IN} = 0.45V$		−800	μA
I_{IH1}	Input high current to RST for reset	$V_{IN} = V_{CC} - 1.5V$		500	μA
I_{LI}	Input leakage current to port 0, $\overline{\text{EA}}$	$0.45 < V_{IN} < V_{CC}$		±10	μA
I_{IL2}	Logical 0 input current for XTAL2	XTAL1 at V_{SS}, $V_{IN} = 0.45V$		−3.2	mA
I_{CC}	Power supply current	All outputs disconnected and $\overline{\text{EA}} = V_{CC}$		175	mA
C_{IO}	Capacitance of I/O buffer	$f_C = 1MHz$, $T_A = 25°C$		10	pF

$T_A = -40°C$ to $+85°C$

SYMBOL	PARAMETER	TEST CONDITIONS	Min	Max	UNIT
V_{IH}	Input high voltage (except RST and XTAL2)		2.2	$V_{CC}+0.5$	V
V_{IH1}	Input high voltage to RST for reset, XTAL2	XTAL1 to V_{SS}	2.7		V
I_{CC}	Power supply current	All outputs disconnected and $\overline{\text{EA}} = V_{CC}$		175	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. For testing, all input signals swing between 0.45V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2V and at output voltages of 0.8V and 2V as appropriate.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. V_{OL} is degraded when the SCN8032AH/SCN8052AH rapidly discharges external capacitance. This AC noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close to the SCN8032AH/SCN8052AH as possible.

Datum	Emitting Ports	Degraded I/O Lines	V_{OL}(Peak Max)
Address	P2, P0	P1, P3	0.8V
Write data	P0	P1, P3, ALE	0.8V

8. $C_L = 100pF$ for port 0, ALE and $\overline{\text{PSEN}}$ outputs; $C_L = 80pF$ for all other ports.

Single-chip 8-Bit Microcontroller

SCN8032AH, SCN8052AH

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 10\%$, $V_{SS} = 0V^{4, 5, 7}$

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK $1/t_{CLCL}$ = 3.5MHz to 12MHz		UNIT
		Min	Max	Min	Max	
Program memory char (Figure 1)						
t_{LHLL}	ALE pulse width	127		$2t_{CLCL} - 40$		ns
t_{AVLL}	Address set-up to ALE	43		$t_{CLCL} - 40$		ns
t_{LLAX}	Address hold after ALE	48		$t_{CLCL} - 35$		ns
t_{LLIV}	ALE to valid instr in		233		$4t_{CLCL} - 100$	ns
t_{LLPL}	ALE to $\overline{\text{PSEN}}$	58		$t_{CLCL} - 25$		ns
t_{PLPH}	$\overline{\text{PSEN}}$ pulse width	215		$3t_{CLCL} - 35$		ns
t_{PLIV}	$\overline{\text{PSEN}}$ to valid instr in		125		$3t_{CLCL} - 125$	ns
t_{PXIX}	Input instr hold after $\overline{\text{PSEN}}$	0		0		ns
t_{PXIZ}	Input instr float after $\overline{\text{PSEN}}$		63		$t_{CLCL} - 20$	ns
t_{PXAV}	Address valid after $\overline{\text{PSEN}}$	75		$t_{CLCL} - 8$		ns
t_{AVIV}	Address to valid instr in		302		$5t_{CLCL} - 115$	ns
t_{PLAZ}	$\overline{\text{PSEN}}$ to address float		20		20	ns
External data memory char (Figures 2 and 3)						
t_{RLRH}	$\overline{\text{RD}}$ pulse width	400		$6t_{CLCL} - 100$		ns
t_{WLWH}	$\overline{\text{WR}}$ pulse width	400		$6t_{CLCL} - 100$		ns
t_{LLAX}	Address hold after ALE	48		$t_{CLCL} - 35$		ns
t_{RLDV}	$\overline{\text{RD}}$ to valid data in		252		$5t_{CLCL} - 165$	ns
t_{RHDX}	Data hold after $\overline{\text{RD}}$	0		0		ns
t_{RHDZ}	Data float after $\overline{\text{RD}}$		97		$2t_{CLCL} - 70$	ns
t_{LLDV}	ALE to valid data in		517		$8t_{CLCL} - 150$	ns
t_{AVDV}	Address to valid data in		585		$9t_{CLCL} - 165$	ns
t_{LLWL}	ALE to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	200	300	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
t_{AVWL}	Address to $\overline{\text{WR}}$ or $\overline{\text{RD}}$	203		$4t_{CLCL} - 130$		ns
t_{WHLH}	$\overline{\text{WR}}$ or $\overline{\text{RD}}$ high to ALE high	43	123	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
t_{DVWX}	Data valid to $\overline{\text{WR}}$ transition	23		$t_{CLCL} - 60$		ns
t_{QVWH}	Data set-up before $\overline{\text{WR}}$	433		$7t_{CLCL} - 150$		ns
t_{WHQX}	Data hold after $\overline{\text{WR}}$	33		$t_{CLCL} - 50$		ns
t_{RLAZ}	Address float after $\overline{\text{RD}}$		20		20	ns
External clock (Figure 4)						
t_{CLCL}	Oscillator period			83.3	286	ns
t_{CHCX}	High time			20		ns
t_{CLCX}	Low time			20		ns
t_{CLCH}	Rise time				20	ns
t_{CHCL}	Fall time				20	ns

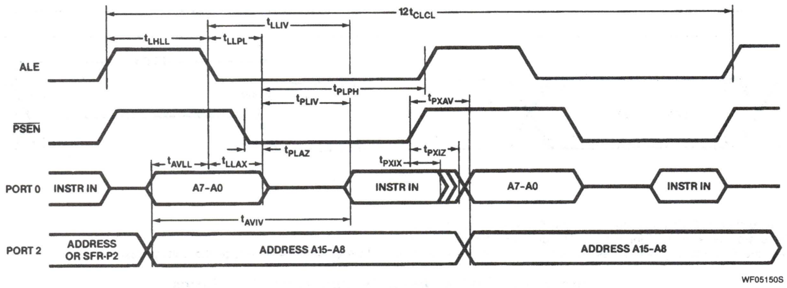


Figure 1. Program Memory Read Cycle

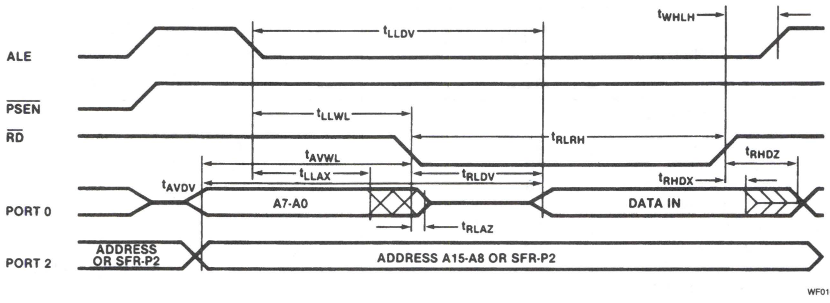

tWHLH
tLLDV
ALE
PSEN
tLLWL
RD
tRLRH
tAVWL
tRHDZ
tLLAX
tRLDV
tRHDX
tAVDV
PORT 0
A7-A0
DATA IN
tRLAZ
PORT 2
ADDRESS OR SFR-P2
ADDRESS A15-A8 OR SFR-P2
WF01630S

Figure 2. Data Memory Read Cycle

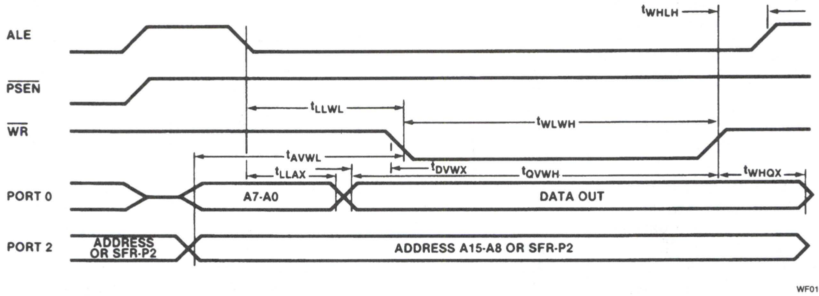

tWHLH
ALE
PSEN
tLLWL
tWLWH
WR
tAVWL
tLLAX
tDVWX
tQVWH
tWHQX
PORT 0
A7-A0
DATA OUT
PORT 2
ADDRESS OR SFR-P2
ADDRESS A15-A8 OR SFR-P2
WF01640S

Figure 3. Data Memory Write Cycle

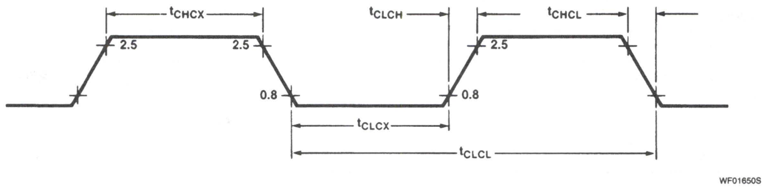
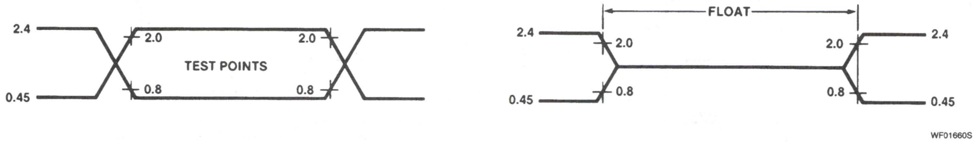
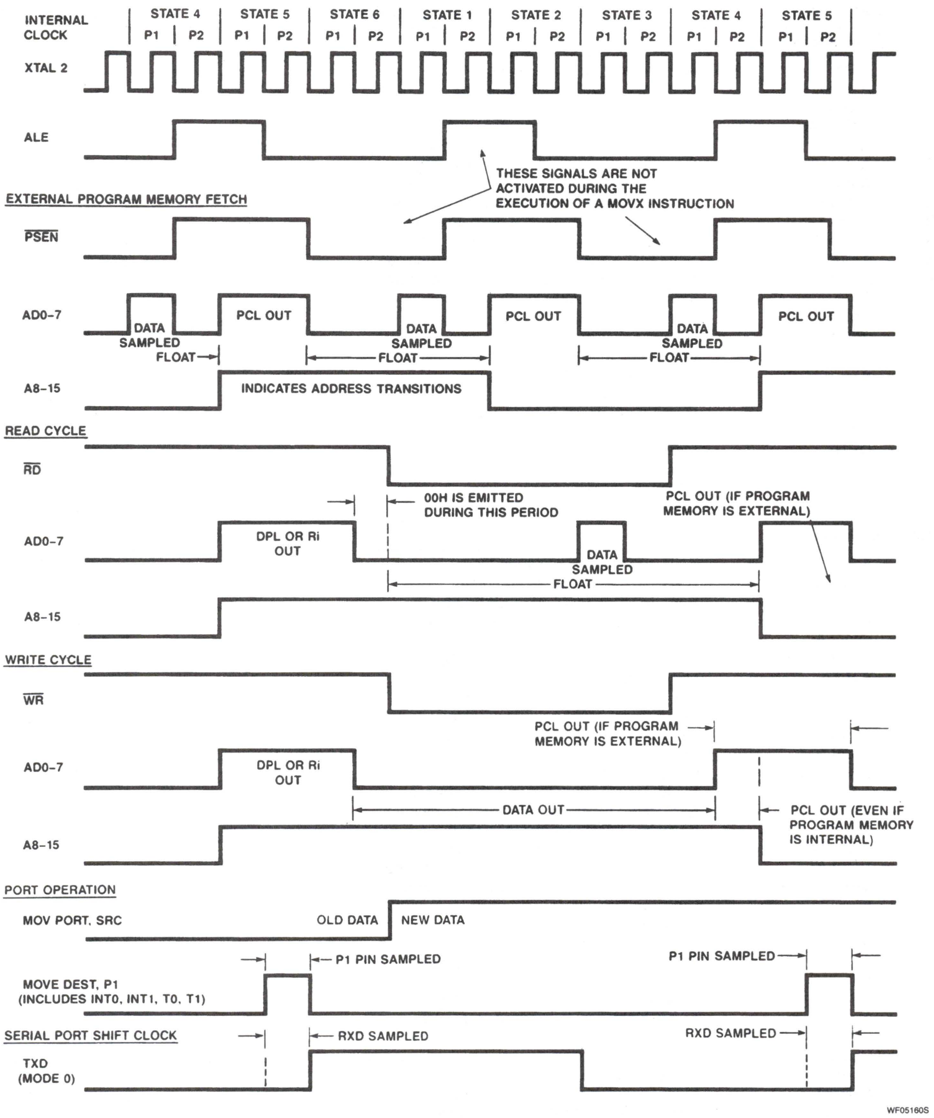

tCHCX
tCLCH
tCHCL
2.5
2.5
2.5
0.8
0.8
tCLCX
tCLCL
WF01650S

Figure 4. External Clock Drive XTAL2


2.4
0.45
2.0
2.0
TEST POINTS
0.8
0.8
FLOAT
2.4
2.0
2.0
2.4
0.45
0.8
0.8
0.45
WF01660S

NOTES:
1. AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0".
2. Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".
3. For timing purposes, the float state is defined as the point at which a P0 pin sinks 2.4mA or sources 400μA at the voltage test levels.

Figure 5. AC Testing Input/Output, Float Waveforms



INTERNAL CLOCK
STATE 4
STATE 5
STATE 6
STATE 1
STATE 2
STATE 3
STATE 4
STATE 5
P1
P2
XTAL 2
ALE
THESE SIGNALS ARE NOT ACTIVATED DURING THE EXECUTION OF A MOVX INSTRUCTION
EXTERNAL PROGRAM MEMORY FETCH
PSEN
AD0-7
DATA SAMPLED
PCL OUT
FLOAT
A8-15
INDICATES ADDRESS TRANSITIONS
READ CYCLE
RD
00H IS EMITTED DURING THIS PERIOD
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL)
AD0-7
DPL OR Ri OUT
DATA SAMPLED
FLOAT
A8-15
WRITE CYCLE
WR
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL)
AD0-7
DPL OR Ri OUT
DATA OUT
PCL OUT (EVEN IF PROGRAM MEMORY IS INTERNAL)
A8-15
PORT OPERATION
MOV PORT, SRC
OLD DATA
NEW DATA
P1 PIN SAMPLED
MOVE DEST, P1 (INCLUDES INT0, INT1, T0, T1)
SERIAL PORT SHIFT CLOCK
RXD SAMPLED
TXD (MODE 0)
WF05160S

NOTE:
All internal timing is referenced to the internal time states shown at the top of the page. This waveform represents the signal on the X2 input of the oscillator. This diagram represents when these signals are actually clocked within the chip. However, the time it takes a signal to propagate to the pins is in the range of 50-150ns. Propagation delays are dependent on many variables, such as temperature and pin loading. Even the different signals vary. Typically though, $\overline{RD}$ and $\overline{WR}$ have propagation delays of approximately 50ns and the other timing signals approximately 85ns. At room temperature, fully loaded, these differences in propagation delays between signals have been integrated into the timing specs.

Figure 6. Timing Waveforms

Table 1. Instruction Set

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Arithmetic Operations				
ADD	A,Rn	Add register to accumulator	1	1
ADD	A,direct	Add direct byte to accumulator	2	1
ADD	A,@Ri	Add indirect RAM to accumulator	1	1
ADD	A,R#data	Add immediate data to accumulator	2	1
ADDC	A,Rn	Add register to accumulator with carry	1	1
ADDC	A,direct	Add direct byte to A with carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1
ADDC	A,#data	Add immediate data to A with carry flag	2	1
SUBB	A,Rn	Subtract register from A with borrow	1	1
SUBB	A,direct	Subtract direct byte from A with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A w/borrow	1	1
SUBB	A,#data	Subtract immed. data from A w/borrow	2	1
INC	A	Increment accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement Indirect RAM	1	1
INC	DPTR	Increment data pointer	1	2
MUL	AB	Mulitiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal adjust accumulator	1	1
Logical Operations				
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate A left through the carry flag	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate A right through carry flag	1	1
SWAP	A	Swap nibbles within the accumulator	1	1
Data Transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct, #data	Move immediate data to direct byte	3	2

Table 1. Instruction Set (Continued)

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Data Transfer (Continued)				
MOV	@Ri,A	Move accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to A	1	2
MOVC	A,@A + PC	Move code byte relative to PC to A	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to external RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order digit ind. RAM w/A	1	1
Boolean Variable Manipulation				
CLR	C	Clear carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry flag	2	2
ANL	C,bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry flag	2	2
ORL	C,bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry flag	2	1
MOV	bit,C	Move carry flag to direct bit	2	2
Program and Machine Control				
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long jump	3	2
SJMP	rel	Short jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if accumulator is zero	2	2
JNZ	rel	Jump if accumulator is not zero	2	2
JC	rel	Jump if carry flag is set	2	2
JNC	rel	Jump if no carry flag	2	2
JB	bit,rel	Jump if direct bit set	3	2
JNB	bit,rel	Jump if direct bit not set	3	2
JBC	bit,rel	Jump if direct bit is set & clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & jump if not equal	3	2
CJNE	A,#data, rel	Comp. immed. to A & jump if not equal	3	2
CJNE	Rn,#data, rel	Comp. immed. to reg. & jump if not equal	3	2
CJNE	@Ri,#data, rel	Comp. immed. to ind. & jump if not equal	3	2
DJNZ	Rn,rel	Decrement register & jump if not zero	2	2
DJNZ	direct,rel	Decrement direct & jump if not zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn	–Working register R0-R7
direct	–128 internal RAM locations, any I/O port, control or status register
@Ri	–Indirect Internal RAM location addressed by register R0 or R1
#data	–8-bit constant included in instruction
#data16	–16-bit constant included as bytes 2 & 3 of instruction
bit	–128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr16	–Destination address for LCALL & LJMP may be anywhere within the 64-kilobyte program memory address space.
addr11	–Destination address for ACALL & AJMP will be within the same 2-kilobyte page of program memory as the first byte of the following instruction.
rel	–SJMP and all conditional jumps include an 8-bit offset byte. Range is +127 – 128 bytes relative to first byte of the following instruction.

Table 2. Instruction Opcodes in Hexadecimal Order

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,#data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr,A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data

Table 2. Instruction Opcodes in Hexadecimal Order (Continued)

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A + PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A + DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Signetics

SC80C31B/SC80C51B
CMOS Single-Chip 8-Bit Microcontroller

Preliminary Specification

Microprocessor Products

DESCRIPTION

The Signetics SC80C31B/SC80C51B is a high-performance microcontroller fabricated with Signetics' high-density CMOS technology. The CMOS SC80C31/51 is functionally compatible with the NMOS SCN8031/51 microcontroller. The Signetics CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Signetics' epitaxial substrate minimizes latch-up sensitivity.

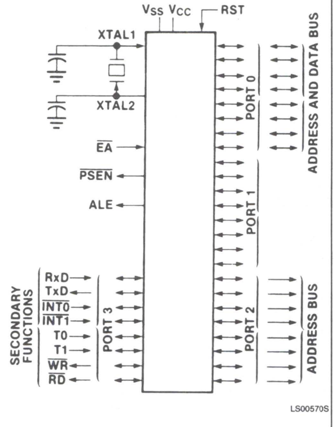
The SC80C31B/SC80C51B contains a 4K × 8 ROM; a 128 × 8 RAM; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority level, nested interrupt structure; a serial I/O port for either multiprocessor communications, I/O expansion or full duplex UART; and on-chip oscillator and clock circuits. In addition, the SC80C31B/80C51B has two software selectable modes of reduced activity for further power reduction – idle mode and power-down mode.

Idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. Power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

FEATURES

- **SC80C31 microcontroller without ROM**
- **SC80C51 control-oriented microcontroller**
- **SC8031/8051 compatible**
  - **4K × 8 ROM**
  - **128 × 8 RAM**
  - **Two 16-bit timer/counters**
  - **Full duplex serial channel**
  - **Boolean processor**
- **Memory addressing capability**
  - **64K ROM and 64K RAM**
- **Power control modes:**
  - **Idle mode**
  - **Power-down mode**
- **CMOS and TTL compatible**
- **Three speed ranges at V_{CC} = 5V ±20%**
  - **3.5 to 12 MHz**
  - **3.5 to 16 MHz**
  - **0.5 to 12 MHz**

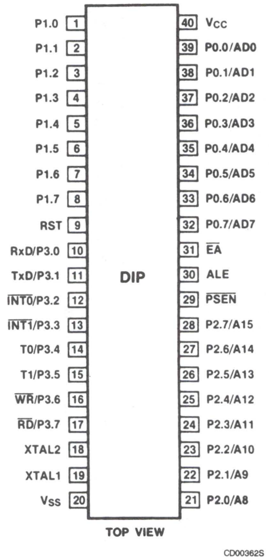
LOGIC SYMBOL



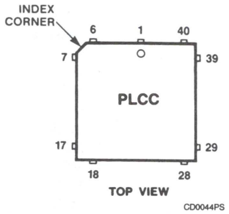
VSS VCC
RST
XTAL1
XTAL2
EA
PSEN
ALE
PORT 0
ADDRESS AND DATA BUS
PORT 1
SECONDARY FUNCTIONS
RxD
TxD
INT0
INT1
T0
T1
WR
RD
PORT 3
PORT 2
ADDRESS BUS
LS00570S

NOTE:
C1, C2 = 30pFd ± 10pFd for crystals
C1, C2 = 40pFd ± 10pFd for ceramic resonators

PIN CONFIGURATIONS



P1.0 1
P1.1 2
P1.2 3
P1.3 4
P1.4 5
P1.5 6
P1.6 7
P1.7 8
RST 9
RxD/P3.0 10
TxD/P3.1 11
INT0/P3.2 12
INT1/P3.3 13
T0/P3.4 14
T1/P3.5 15
WR/P3.6 16
RD/P3.7 17
XTAL2 18
XTAL1 19
Vss 20
40 Vcc
39 P0.0/AD0
38 P0.1/AD1
37 P0.2/AD2
36 P0.3/AD3
35 P0.4/AD4
34 P0.5/AD5
33 P0.6/AD6
32 P0.7/AD7
31 EA
30 ALE
29 PSEN
28 P2.7/A15
27 P2.6/A14
26 P2.5/A13
25 P2.4/A12
24 P2.3/A11
23 P2.2/A10
22 P2.1/A9
21 P2.0/A8
DIP
TOP VIEW
CD00362S



INDEX CORNER
6
1
40
7
39
PLCC
17
29
18
28
TOP VIEW
CD0044PS

Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0	24	A8/P2.0
3	P1.1	25	A9/P2.1
4	P1.2	26	A10/P2.2
5	P1.3	27	A11/P2.3
6	P1.4	28	A12/P2.4
7	P1.5	29	A13/P2.5
8	P1.6	30	A14/P2.6
9	P1.7	31	A15/P2.7
10	RST	32	PSEN
11	RxD/P3.0	33	ALE
12	NC	34	NC
13	TxD/P3.1	35	EA
14	INT0/P3.2	36	P0.7/AD7
15	INT1/P3.3	37	P0.6/AD6
16	T0/P3.4	38	P0.5/AD5
17	T1/P3.5	39	P0.4/AD4
18	WR/P3.6	40	P0.3/AD3
19	RD/P3.7	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{CC}

ORDERING INFORMATION

SC80C□□ B □ □ □ □□ (CPxxxx)

ROM/RAM (bytes)
31 = EXT/128
51 = 4K/128

CUSTOM ROM PATTERN NUMBER
Applies to masked ROM versions only. Number will be assigned by Signetics. Contact Signetics sales office for ROM pattern submission requirements.

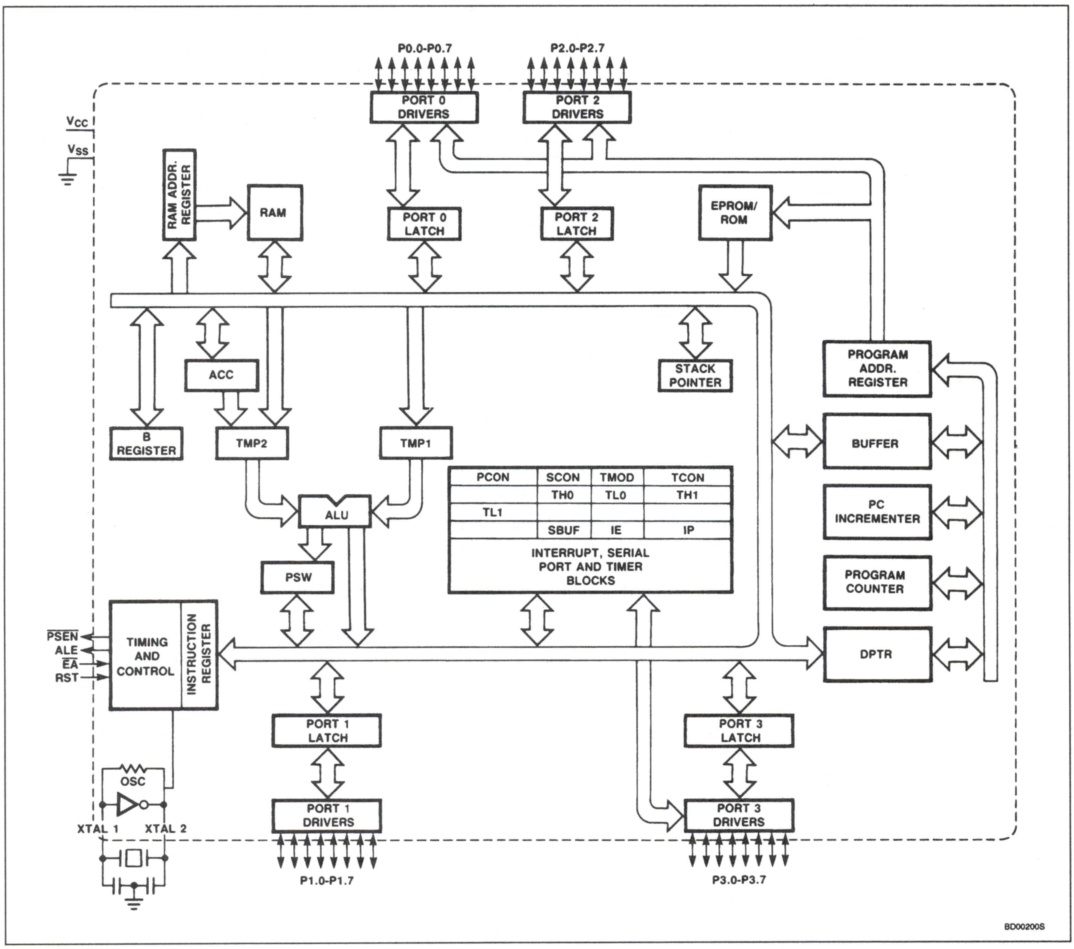
OPERATING TEMPERATURE RANGE
A = 40°C to +85°C
C = 0°C to +70°C

40=Pin DIP
44=Pin LCC

SPEED
0 = 3.5 to 12MHz
1 = 3.5 to 16MHz
2 = 0.5 to 12MHz

PACKAGE
N = Plastic DIP
I = Ceramic DIP
A = Plastic LCC

BLOCK DIAGRAM


P0.0-P0.7
P2.0-P2.7
PORT 0 DRIVERS
PORT 2 DRIVERS
Vcc
Vss
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
EPROM/ ROM
ACC
STACK POINTER
PROGRAM ADDR. REGISTER
B REGISTER
TMP2
TMP1
BUFFER
PCON
SCON
TMOD
TCON
TH0
TL0
TH1
TL1
SBUF
IE
IP
INTERRUPT, SERIAL PORT AND TIMER BLOCKS
ALU
PC INCREMENTER
PSW
PROGRAM COUNTER
PSEN
ALE
EA
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
PORT 1 LATCH
PORT 3 LATCH
OSC
XTAL 1
XTAL 2
PORT 1 DRIVERS
PORT 3 DRIVERS
P1.0-P1.7
P3.0-P3.7
BD00200S

CMOS Single-Chip 8-Bit Microcontroller SC80C31B/SC80C51B

PIN DESCRIPTION

MNEMONIC	PIN NO. DIP	PIN NO. LCC	TYPE	NAME AND FUNCTION
V_{SS}	20	22		Circuit ground potential.
V_{CC}	40	44		Supply voltage during normal, idle, and power-down operation.
P0.0–P0.7	39–32	43–36	I/O	**Port 0:** Port 0 is an 8-bit open drain, bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s. Port 0 also outputs the code bytes during program verification in the 80C51BH. External pullups are required during program verification.
P1.0–P1.7	1–8	2–9	I/O	**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are eternally being pulled low will source current (IIL, on the data sheet) because of the internal pullups. Port 1 also receives the low-order address byte during program verification.
P2.0–P2.7	21–18	24–31	I/O	**Port 2:** Port 2 Is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups. Port 2 emits the high-order address byte during fetches from external Program Memory and during access to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.
P3.0–P3.7	10–17	11, 13–19	I/O	**Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. Port 3 also serves the functions of various special features of the SC80C51 Family, as listed below:
	10	11	I	RXD (P3.0): Serial input port
	11	13	O	TXD (P3.1): Serial output port
	12	14	I	$\overline{\text{INT0}}$ (P3.2): External interrupt
	13	15	I	$\overline{\text{INT1}}$ (P3.3): External interrupt
	14	16	I	T0 (P3.4): Timer 0 external input
	15	17	I	T1 (P3.5): Timer 1 external input
	16	18	O	$\overline{\text{WR}}$ (P3.6): External data memory write strobe
	17	19	O	$\overline{\text{RD}}$ (P3.7): External data memory read strobe.
RST	9	10	I	**Reset:** A high on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits Power-On reset using only an external capacitor V_{CC}.
ALE	30	33	O	**Address Latch Enable:** Output pulse for latching the low byte of the address during accesses to external memory. In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.
$\overline{\text{PSEN}}$	29	32	O	**Program Store Enable:** Is the read strobe to external Program Memory. When the 80C51BH is executing code from external Program Memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external Data Memory. $\overline{\text{PSEN}}$ is not activated during fetches from internal program memory.
$\overline{\text{EA}}$	31	35	I	**External Access Enable:** $\overline{\text{EA}}$ must be externally held low in order to enable to device to fetch code from external Program Memory locations 0000H to 0FFFFH. If $\overline{\text{EA}}$ is held high the device executes from internal Program Memory unless the program counter contains an address greater than 0FFFH.
XTAL1	19	21	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	O	**Crystal 2:** Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in the Logic Symbol on page 1.

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

IDLE MODE

In the idle mode, the CPU puts itself to sleep while all the on chip peripherals stay active. The instruction that invokes the Idle mode is the last instruction executed in the normal operating mode before Idle mode is activated. The content of CPU, the on chip RAM, and all the Special Function Registers remain intact during this mode. The Idle mode can be terminated either by any enabled interrupt, at which time the process is picked up at the interrupt service routine and continued, or by a hardware reset which starts the processor the same as a power on reset.

POWER DOWN MODE

In the Power Down mode the oscillator is stopped, and the instruction that invokes Power Down is the last instruction executed. Only the contents of the on chip RAM is preserved. A hardware reset is the only way to terminate Power Down.

The control bits for the reduced power modes are in the Special Function Register PCON.

Table 1. Status of the External Pins During Idle and Power Down Modes

MODE	PROGRAM MEMORY	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	External	0	0	Float	Data	Data	Data
Power Down	Internal	0	0	Data	Data	Data	Data

CMOS Single-Chip 8-Bit Microcontroller

SC80C31B/SC80C51B

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature range	0 to +70	°C
T_{STG}	Storage temperature range	−65 to +150	°C
	Voltage from V_{CC} to V_{SS}[2]	−0.5 to +6.5	V
	Voltage from any pin to V_{SS}[2]	−0.5 to V_{CC}+0.5	V
P_D	Power dissipation	1	W

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 4V$ to $6V$, $V_{SS} = 0V$[3]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{IL}	Input low voltage, except $\overline{EA}$		−0.5	$0.2V_{CC}$−0.1	V
V_{IL1}	Input low voltage, $\overline{EA}$		−0.5	$0.2V_{CC}$−0.3	V
V_{IH}	Input high voltage except XTAL1, RST		$0.2V_{CC}$+0.9	V_{CC}+0.5	V
V_{IH1}	Input high voltage, XTAL1, RST		$0.7V_{CC}$	V_{CC}+0.5	V
V_{OL}	Output low voltage, ports 1, 2, 3	$I_{OL} = 1.6mA$[4]		0.45	V
V_{OL1}	Output low voltage, port 0, ALE, $\overline{PSEN}$	$I_{OL} = 3.2mA$[4]		0.45	V
V_{OH}	Output high voltage, ports 1, 2, 3	$I_{OH} = -60\mu A$, $V_{CC} = 5V \pm 10\%$	2.4		V
		$I_{OH} = -25\mu A$	$0.75V_{CC}$		V
		$I_{OH} = -10\mu A$	$0.9V_{CC}$		V
V_{OH1}	Output high voltage, port 0 in external bus mode, ALE, $\overline{PSEN}$	$I_{OH} = -400\mu A$, $V_{CC} = 5V \pm 10\%$	2.4		V
		$I_{OH} = -150\mu A$	$0.75V_{CC}$		V
		$I_{OH} = -40\mu A$[5]	$0.9V_{CC}$		V
I_{IL}	Logical 0 input current, ports 1, 2, 3	$V_{IN} = 0.45V$		−50	µA
I_{TL}	Logical 1 to 0 transition current, ports 1, 2, 3	$V_{IN} = 2V$		−650	µA
I_{LI}	Input leakage, current port 0, $\overline{EA}$	$0.45 < V_{in} < V_{CC}$		±10	µA
R_{RST}	RST pulldown resistor		50	150	kΩ
C_{IO}	Pin capacitance	Test freq = 1MHz, $T_A = 25°C$		10	pF
I_{PD}	Power-down current	$V_{CC} = 2$ to $6V$[6]		50	µA

MAXIMUM I_{CC}(mA)

	OPERATING (NOTE 7)			IDLE (NOTE 8)		
Freq. V_{CC}	4V	5V	6V	4V	5V	6V
0.5MHz	1.6	2.2	3	0.6	0.9	1.2
3.5MHz	4.3	5.7	7.5	1.1	1.6	2.2
8.0MHz	8.3	11	14	1.8	2.7	3.7
12MHz	12	16	20	2.5	3.7	5
16MHz	16	20.5	25	3.5	5	6.5

TYPICAL I_{CC}: Typical operating I_{CC} at 3 MHz and $V_{CC} = 5.0V$ is 3.8 mA; and varies with frequency at the rate of 0.67 mA/MHz.

AC ELECTRICAL CHARACTERISTICS

$T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 20\%$, $V_{SS} = 0V$, load capacitance for port 0, ALE, and $\overline{PSEN} = 100pF$, load capacitance for all other outputs = 80pF[3].

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator frequency					MHz
	SC80C451B□0/SC83C451B□0			3.5	12	
	SC80C451B□1/SC83C451B□1			3.5	16	
	SC80C451B□2/SC83C451B□2			0.5	12	
t_{LHLL}	ALE pulse width	127		$2t_{CLCL}-40$		ns
t_{AVLL}	Address valid to ALE Low	28		$t_{CLCL}-55$		ns
t_{LLAX}	Address hold after ALE Low	48		$t_{CLCL}-35$		ns
t_{LLIV}	ALE low to valid instr in		234		$4t_{CLCL}-100$	ns
t_{LLPL}	ALE low to $\overline{PSEN}$ Low	43		$t_{CLCL}-40$		ns
t_{PLPH}	$\overline{PSEN}$ pulse width	205		$3t_{CLCL}-45$		ns
t_{PLIV}	$\overline{PSEN}$ low to valid instr in		145		$3t_{CLCL}-105$	ns
t_{PXIX}	Input instr hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input instr float after $\overline{PSEN}$		59		$t_{CLCL}-25$	ns
t_{AVIV}	Address to valid instrument in		312		$5t_{CLCL}-105$	ns
t_{PLAZ}	$\overline{PSEN}$ low to address float		10		10	ns
t_{RLRH}	$\overline{RD}$ pulse width	400		$6t_{CLCL}-100$		ns
t_{WLWH}	$\overline{WR}$ pulse width	400		$6t_{CLCL}-100$		ns
t_{RLDV}	$\overline{RD}$ low to valid data in		252		$5t_{CLCL}-165$	ns
t_{RHDX}	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data float after $\overline{RD}$		97		$2t_{CLCL}-70$	ns
t_{LLDV}	ALE low to valid data in		517		$8t_{CLCL}-150$	ns
t_{AVDV}	Address to valid data in		585		$9t_{CLCL}-165$	ns
t_{LLWL}	ALE low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address valid to write Low	203		$4t_{CLCL}-30$		ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	23		$t_{CLCL}-60$		ns
t_{WHQX}	Data hold after $\overline{WR}$	33		$t_{CLCL}-50$		ns
t_{RLAZ}	$\overline{RD}$ low to address float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ high to ALE High	43	123	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
t_{CHCX}	High Time	20				ns
t_{CLCX}	Low Time	20				ns
t_{CLCH}	Rise Time		20			ns
t_{CHCL}	Fall Time		20			ns
t_{XLXL}	Serial Port Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10T_{CLCL}-133$		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-117$		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 't' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address.
C: Clock.
D: Input data.
H: Logic level High.
I: Instruction (program memory contents).
L: Logic level Low, or ALE.
P: PSEN.
Q: Output data.
R: READ signal.
t: Time.
V: Valid.
W: WRITE signal.
X: No longer a valid logic level.
Z: Float.

EXAMPLE:

t_{AVLL} = Time for Address Valid to ALE Low.

t_{LLPL} = Time for ALE Low to $\overline{PSEN}$ Low.

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maximum.
3. Parameters are valid over operating temperature range unless otherwise specified.
4. Capacitive loading on ports 0 and 2 can cause spurious noise pulses to be superimposed on the V_{OLS} of ALE and ports 1 and 3. The noise is caused by external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE line can exceed 0.8V. In such cases it could be desirable to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger STROBE input.
5. Capacitive loading on ports 0 and 2 can cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall before the 0.9 V_{CC} specification when the address bits are stabilizing.
6. Power-down I_{CC} is measured with all output pins disconnected; $\overline{EA}$ = port 0 = V_{CC}; XTAL2 N.C.; RST = V_{SS}.
7. I_{CC} is measured with all output pins disconnected; XTAL1 driven with t_{CLCH}, t_{CHCL} = 5ns, $V_{IL} = V_{SS}$ +.05V, $V_{IH} = V_{CC}$ −0.5V; XTAL2 N.C.; $\overline{EA}$ = RST = Port 0 = V_{CC}. I_{CC} will be slightly higher if a crystal oscillator is used.
8. Idle I_{CC} is measured with all output pins disconnected; XTAL1 driven with t_{CLCH}, t_{CHCL} = 5ns, $V_{IL} = V_{SS}$ + 0.5V, $V_{IH} = V_{CC}$ −0.5V; XTAL2 N.C.; port 0 = V_{CC}; $\overline{EA}$ = RST = V_{SS}.

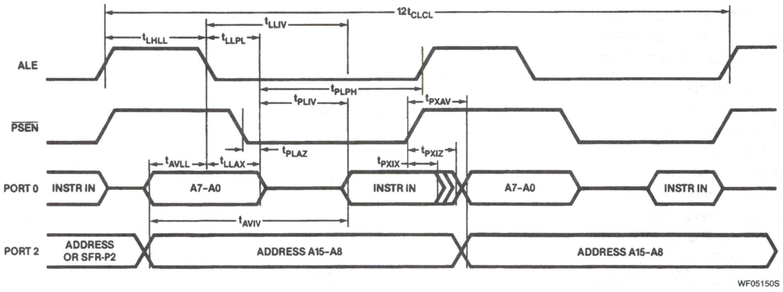


Figure 1. Program Memory Read Cycle

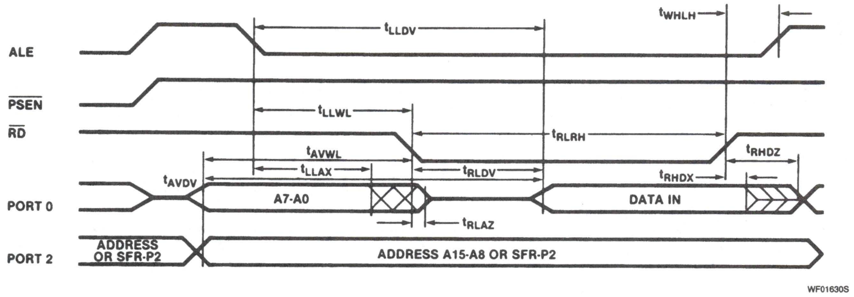

ALE
PSEN
RD
PORT 0
PORT 2
t_{LLDV}
t_{WHLH}
t_{LLWL}
t_{RLRH}
t_{AVWL}
t_{LLAX}
t_{RLDV}
t_{RHDZ}
t_{AVDV}
t_{RHDX}
A7-A0
DATA IN
t_{RLAZ}
ADDRESS OR SFR-P2
ADDRESS A15-A8 OR SFR-P2
WF01630S

Figure 2. Data Memory Read Cycle

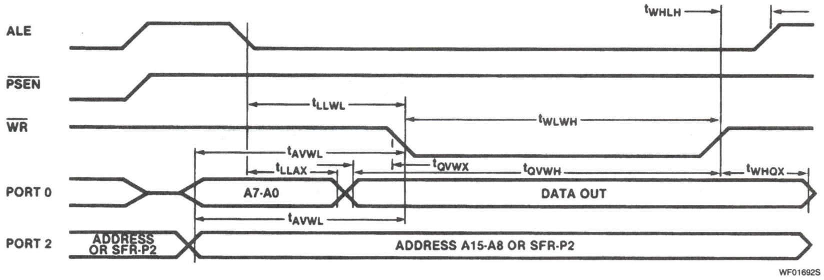

ALE
PSEN
WR
PORT 0
PORT 2
t_{WHLH}
t_{LLWL}
t_{WLWH}
t_{AVWL}
t_{LLAX}
t_{QVWX}
t_{QVWH}
t_{WHQX}
A7-A0
DATA OUT
t_{AVWL}
ADDRESS OR SFR-P2
ADDRESS A15-A8 OR SFR-P2
WF01692S

Figure 3. Data Memory Write Cycle

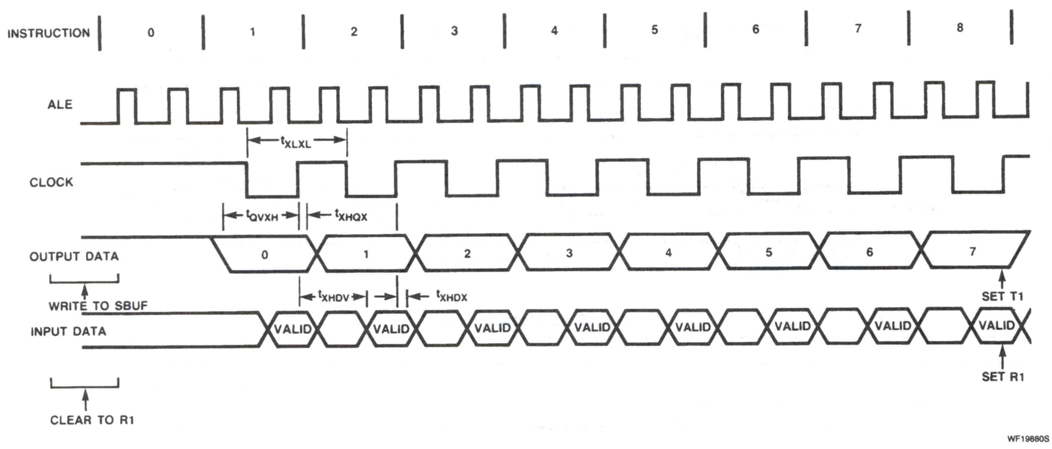

INSTRUCTION
0
1
2
3
4
5
6
7
8
ALE
t_{XLXL}
CLOCK
t_{QVXH}
t_{XHQX}
OUTPUT DATA
0
1
2
3
4
5
6
7
WRITE TO SBUF
t_{XHDV}
t_{XHDX}
INPUT DATA
VALID
SET T1
SET R1
CLEAR TO R1
WF19880S

Figure 4. Shift Register Mode Timing Waveforms

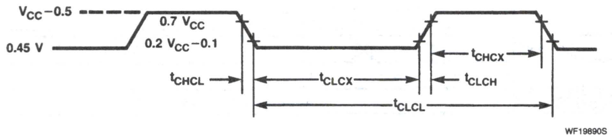

$V_{CC}-0.5$
0.45 V
0.7 V_{CC}
0.2 $V_{CC}-0.1$
t_{CHCL}
t_{CLCX}
t_{CHCX}
t_{CLCH}
t_{CLCL}
WF19890S

Figure 5. External Clock Drive Waveform

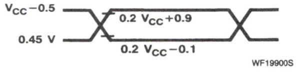

$V_{CC}-0.5$
0.45 V
0.2 $V_{CC}+0.9$
0.2 $V_{CC}-0.1$
WF19900S

Figure 6. AC Testing Input, Output Waveforms[1]

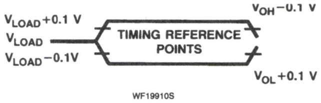
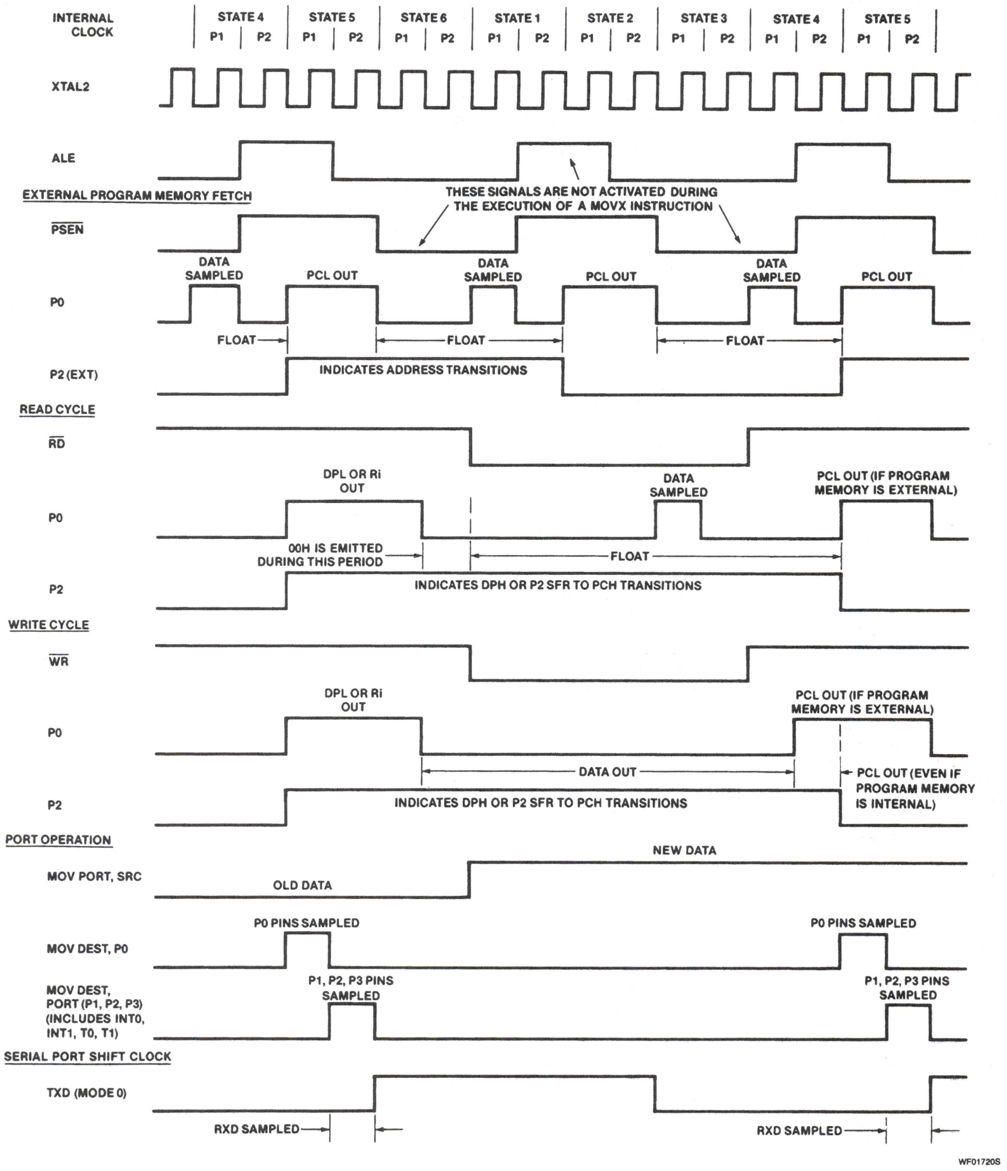

$V_{LOAD}+0.1$ V
V_{LOAD}
$V_{LOAD}-0.1$V
TIMING REFERENCE POINTS
$V_{OH}-0.1$ V
$V_{OL}+0.1$ V
WF19910S

Figure 7. Float Waveforms[2]

NOTES:

1. AC inputs during testing are driven at V_{CC}–0.5 for a logic "1" and 0.45V for a logic "0". Timing measurements are made at V_{IH} min. for a logic "1" and V_{IL} max. for a logic "0".
2. For timing purposes a port pin is no longer floating when a 100mV change from load voltage occurs, and begins to flot when a 100mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} > \pm 20$mA.


INTERNAL CLOCK
STATE 4
STATE 5
STATE 6
STATE 1
STATE 2
STATE 3
STATE 4
STATE 5
P1
P2
XTAL2
ALE
EXTERNAL PROGRAM MEMORY FETCH
THESE SIGNALS ARE NOT ACTIVATED DURING THE EXECUTION OF A MOVX INSTRUCTION
PSEN
DATA SAMPLED
PCL OUT
P0
FLOAT
P2 (EXT)
INDICATES ADDRESS TRANSITIONS
READ CYCLE
RD
DPL OR Ri OUT
DATA SAMPLED
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL)
P0
00H IS EMITTED DURING THIS PERIOD
FLOAT
P2
INDICATES DPH OR P2 SFR TO PCH TRANSITIONS
WRITE CYCLE
WR
DPL OR Ri OUT
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL)
P0
DATA OUT
PCL OUT (EVEN IF PROGRAM MEMORY IS INTERNAL)
P2
INDICATES DPH OR P2 SFR TO PCH TRANSITIONS
PORT OPERATION
NEW DATA
MOV PORT, SRC
OLD DATA
P0 PINS SAMPLED
MOV DEST, P0
P0 PINS SAMPLED
P1, P2, P3 PINS SAMPLED
MOV DEST, PORT (P1, P2, P3) (INCLUDES INT0, INT1, T0, T1)
P1, P2, P3 PINS SAMPLED
SERIAL PORT SHIFT CLOCK
TXD (MODE 0)
RXD SAMPLED
RXD SAMPLED
WF01720S

NOTE:
This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25ns to 125ns. This propagation is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component to component. Typically though (T_A = 25°C, fully loaded), $\overline{RD}$ and $\overline{WR}$ propagation delays are approximately 50ns. The other signals are typically 85ns. Propagation delays are incorporated in the AC specifications.

Figure 8. Timing Waveforms

Table 2. Instruction Set

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Arithmetic operations				
ADD	A,Rn	Add register to accumulator	1	1
ADD	A,direct	Add direct byte to accumulator	2	1
ADD	A,@Ri	Add indirect RAM to accumulator	1	1
ADD	A,R#data	Add immediate data to accumulator	2	1
ADDC	A,Rn	Add register to accumulator with carry	1	1
ADDC	A,direct	Add direct byte to A with carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1
ADDC	A,#data	Add immediate data to A with carry flag	2	1
SUBB	A,Rn	Subtract register from A with borrow	1	1
SUBB	A,direct	Subtract direct byte from A with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A w/borrow	1	1
SUBB	A,#data	Subtract immediate data from A w/borrow	2	1
INC	A	Increment accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement Indirect RAM	1	1
INC	DPTR	Increment data pointer	1	2
MUL	AB	Mulitiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal adjust accumulator	1	1
Logical operations				
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate A left through the carry flag	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate A right through carry flag	1	1
SWAP	A	Swap nibbles within the accumulator	1	1
Data transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move accumulator to indirect RAM	1	1

CMOS Single-Chip 8-Bit Microcontroller SC80C31B/SC80C51B

Table 2. Instruction Set (Continued)

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Data transfer (continued)				
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to A	1	2
MOVC	A,@A + PC	Move code byte relative to PC to A	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to external RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order digit ind. RAM w/A	1	1
Boolean variable manipulation				
CLR	C	Clear carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry flag	2	2
ANL	C,/bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry flag	2	2
ORL	C,/bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry flag	2	1
MOV	bit,C	Move carry flag to direct bit	2	2
Program and machine control				
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long jump	3	2
SJMP	rel	Short jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if accumulator is zero	2	2
JNZ	rel	Jump if accumulator is not zero	2	2
JC	rel	Jump if carry flag is set	2	2
JNC	rel	Jump if no carry flag	2	2
JB	bit,rel	Jump if direct bit set	3	2
JNB	bit,rel	Jump if direct bit not set	3	2
JBC	bit,rel	Jump if direct bit is set & clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & jump if not equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & jump if not equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg. & jump if not equal	3	2
CJNE	@Ri,#data,rel	Comp. immed. to ind. & jump if not equal	3	2
DJNZ	Rn,rel	Decrement register & jump if not zero	2	2
DJNZ	direct,rel	Decrement direct & jump if not zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn	– Working register R0-R7
direct	– 128 internal RAM locations, any I/O port, control or status register
@Ri	– Indirect Internal RAM location addressed by register R0 or R1
#data	– 8-bit constant included in instruction
#data16	– 16-bit constant included as bytes 2 & 3 of instruction
bit	– 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr16	– Destination address for LCALL & LJMP may be anywhere within the 64-kilobyte program memory address space.
addr11	– Destination address for ACALL & AJMP will be within the same 2-kilobyte page of program memory as the first byte of the following instruction.
rel	– SJMP and all conditional jumps include an 8-bit offset byte. Range is +127 – 128 bytes relative to first byte of the following instruction.

Table 3. Instruction Opcodes in Hexadecimal Order

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,#data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr,A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data

Table 3. Instruction Opcodes in Hexadecimal Order (Continued)

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A + PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A + DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Signetics

Microprocessor Products

SC80C451/SC83C451
I/O Expanded CMOS Microcontroller

Preliminary Specification

DESCRIPTION

The Signetics SC80C451/SC83C451 is an I/O expanded, single-chip microcontroller fabricated with Signetics' high-density epitaxial CMOS technology. It is offered in a 68-pin PLCC and a 64-pin DIP. The Signetics CMOS technology combines the high-speed and high-density characteristics of HMOS with the low power of CMOS. Signetics' epitaxial substrate minimizes latch-up sensitivity.

The SC80C451/SC83C451 is a functional extension of the SC80C31/SC80C51 microcontroller with three additional I/O ports and four I/O control lines. The PLCC version has a total of 68 pins. The DIP version has a total of 64 pins. The four added control lines associated with port 6 facilitate high-speed asynchronous I/O functions.

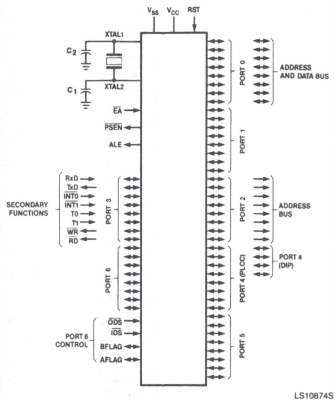
The SC80C451/SC83C451 includes a 4k × 8 ROM; a 128 × 8 RAM; 56 (PLCC) or 52 (DIP) I/O lines; two 16-bit timer/counters; a five source, two priority level, nested interrupt structure; a serial I/O port for either a full-duplex UART, I/O expansion, or multiprocessor communications; and on-chip oscillator and clock circuits.

The SC80C451/SC83C451 has two software selectable modes of reduced activity for further power reduction; idle mode and power-down mode. Idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. Power-down mode freezes the oscillator, causing all other chip functions to be inoperative while maintaining the RAM contents.

FEATURES

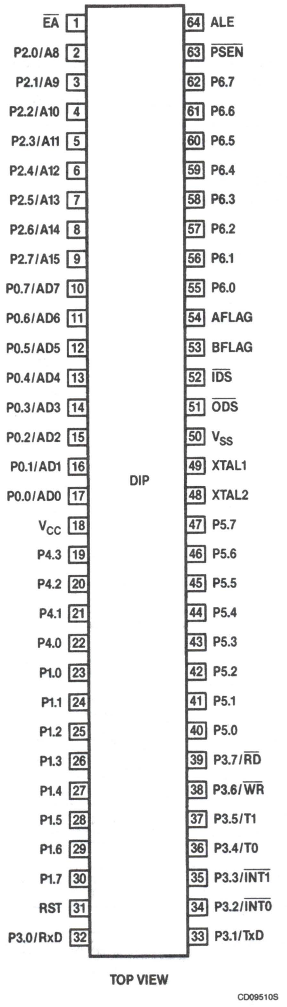
- **SC80C451 — CPU with RAM and I/O**
- **SC83C451 — 80C451 with mask-programmable ROM**
- **68-pin PLCC and 64-pin DIP packages**
  - **Seven 8-bit I/O ports (PLCC version)**
  - **Six 8-bit ports and one 4-bit port (DIP version)**
- **Port 6 features:**
  - **8 data pins**
  - **4 control pins**
  - **Direct MPU bus interface**
  - **Parallel printer interface**
- **Pin compatible with SC87C451 microcontroller**
- **On the microcontroller:**
  - **SC80C51 CPU**
  - **4k × 8 ROM (SC83C451 only)**
  - **128 × 8 RAM**
  - **Two 16-bit timer/counters**
  - **Full-duplex serial I/O**
  - **Two external interrupts**
- **External memory addressing capability:**
  - **64k ROM and 64k RAM**
- **Low power consumption:**
  - **Normal operation: 24mA typical @ 5V, 12MHz**
  - **Idle mode: 3mA typical @ 5V, 12MHz**
  - **Power-down mode: 50μA typical @ 2V**

LOGIC SYMBOL


V_{SS}
V_{CC}
RST
XTAL1
XTAL2
C_2
C_1
$\overline{EA}$
$\overline{PSEN}$
ALE
SECONDARY FUNCTIONS
RxD
TxD
$\overline{INT0}$
$\overline{INT1}$
T0
T1
$\overline{WR}$
$\overline{RD}$
PORT 3
PORT 6
PORT 6 CONTROL
$\overline{ODS}$
$\overline{IDS}$
BFLAG
AFLAG
PORT 0
ADDRESS AND DATA BUS
PORT 1
PORT 2
ADDRESS BUS
PORT 4 (PLCC)
PORT 4 (DIP)
PORT 5
LS10874S

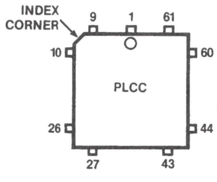
NOTES:
C1, C2 = 30pfd± 10pfd for crystals
= 40pfd± 10pfd for ceramic resonators

PIN CONFIGURATIONS


$\overline{EA}$ 1
P2.0/A8 2
P2.1/A9 3
P2.2/A10 4
P2.3/A11 5
P2.4/A12 6
P2.5/A13 7
P2.6/A14 8
P2.7/A15 9
P0.7/AD7 10
P0.6/AD6 11
P0.5/AD5 12
P0.4/AD4 13
P0.3/AD3 14
P0.2/AD2 15
P0.1/AD1 16
P0.0/AD0 17
V_{CC} 18
P4.3 19
P4.2 20
P4.1 21
P4.0 22
P1.0 23
P1.1 24
P1.2 25
P1.3 26
P1.4 27
P1.5 28
P1.6 29
P1.7 30
RST 31
P3.0/RxD 32
64 ALE
63 $\overline{PSEN}$
62 P6.7
61 P6.6
60 P6.5
59 P6.4
58 P6.3
57 P6.2
56 P6.1
55 P6.0
54 AFLAG
53 BFLAG
52 $\overline{IDS}$
51 $\overline{ODS}$
50 V_{SS}
49 XTAL1
48 XTAL2
47 P5.7
46 P5.6
45 P5.5
44 P5.4
43 P5.3
42 P5.2
41 P5.1
40 P5.0
39 P3.7/$\overline{RD}$
38 P3.6/$\overline{WR}$
37 P3.5/T1
36 P3.4/T0
35 P3.3/$\overline{INT1}$
34 P3.2/$\overline{INT0}$
33 P3.1/TxD
DIP

TOP VIEW

CD09510S


INDEX CORNER
9
1
61
10
60
PLCC
26
44
27
43

TOP VIEW

CD09520S

Pin	Function	Pin	Function
1	$\overline{EA}$	35	RST
2	P2.0/A8	36	P3.0/R × D
3	P2.1/A9	37	P3.1/T × D
4	P2.2/A10	38	P3.2/$\overline{INT0}$
5	P2.3/A11	39	P3.3/$\overline{INT1}$
6	P2.4/A12	40	P3.4/T0
7	P2.5/A13	41	P3.5/T1
8	P2.6/A14	42	P3.6/$\overline{WR}$
9	P2.7/A15	43	P3.7/$\overline{RD}$
10	P0.7/AD7	44	P5.0
11	P0.6/AD6	45	P5.1
12	P0.5/AD5	46	P5.2
13	P0.4/AD4	47	P5.3
14	P0.3/AD3	48	P5.4
15	P0.2/AD2	49	P5.5
16	P0.1/AD1	50	P5.6
17	P0.0/AD0	51	P5.7
18	V_{CC}	52	XTAL2
19	P4.7	53	XTAL1
20	P4.6	54	V_{SS}
21	P4.5	55	$\overline{ODS}$
22	P4.4	56	$\overline{IDS}$
23	P4.3	57	BFLAG
24	P4.2	58	AFLAG
25	P4.1	59	P6.0
26	P4.0	60	P6.1
27	P1.0	61	P6.2
28	P1.1	62	P6.3
29	P1.2	63	P6.4
30	P1.3	64	P6.5
31	P1.4	65	P6.6
32	P1.5	66	P6.7
33	P1.6	67	$\overline{PSEN}$
34	P1.7	68	ALE

I/O Expanded CMOS Microcontroller SC80C451/SC83C451

ORDERING INFORMATION

SC□□ C451 A□ □ □ 6□ (CPxxxx)

ROM/RAM (bytes)
80 = EXT/128
83 = 4k/128

OPERATING TEMPERATURE RANGE
A = –40°C to +85°C
C = –0°C to +70°C

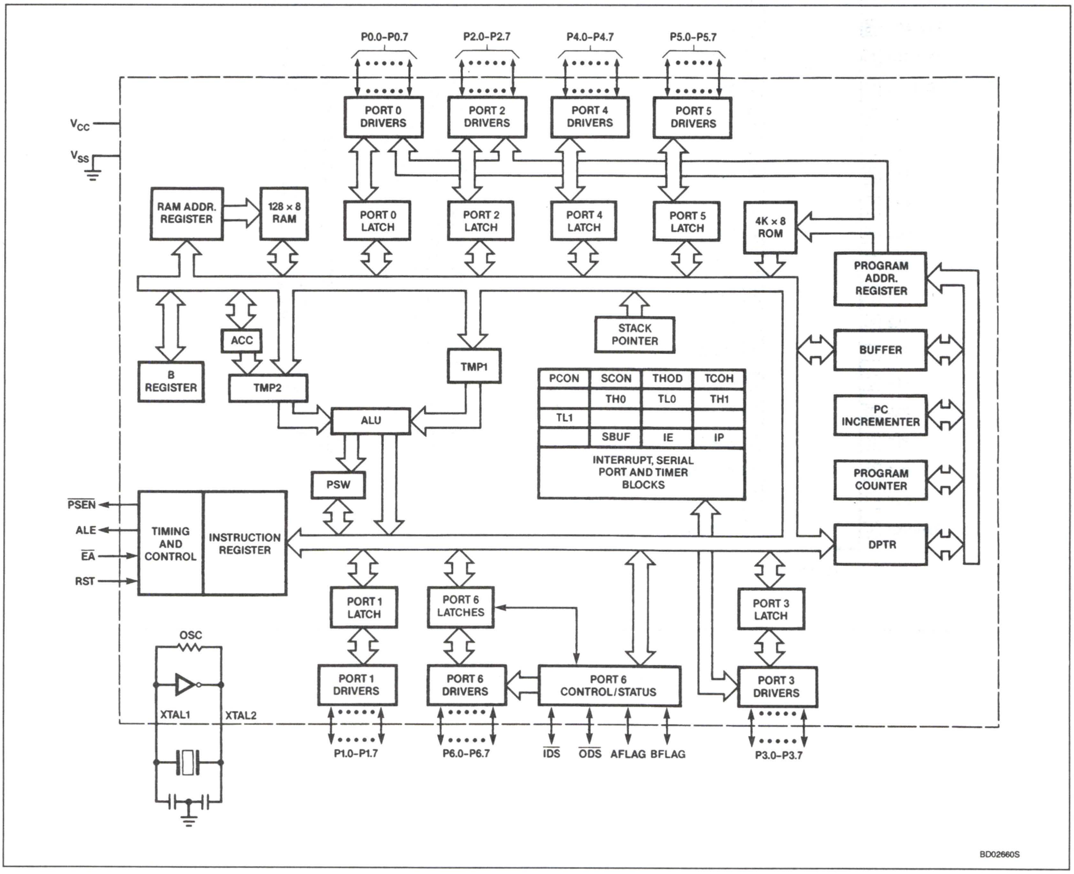
SPEED
0 = 3.5 to 12MHz
1 = 3.5 to 16MHz
2 = 0.5 to 12MHz

CUSTOM ROM PATTERN NUMBER
Applies to masked ROM versions only. Number will be assigned by Signetics. Contact Signetics sales office for ROM pattern submission requirements.

68-Pin LCC
64-Pin DIP

PACKAGE
N = Plastic DIP
I = Ceramic DIP
A = Plastic LCC

BLOCK DIAGRAM


P0.0–P0.7
P2.0–P2.7
P4.0–P4.7
P5.0–P5.7
VCC
VSS
PORT 0 DRIVERS
PORT 2 DRIVERS
PORT 4 DRIVERS
PORT 5 DRIVERS
RAM ADDR. REGISTER
128 x 8 RAM
PORT 0 LATCH
PORT 2 LATCH
PORT 4 LATCH
PORT 5 LATCH
4K x 8 ROM
PROGRAM ADDR. REGISTER
STACK POINTER
ACC
B REGISTER
TMP2
TMP1
BUFFER
PCON
SCON
THOD
TCOH
THO
TLO
TH1
TL1
SBUF
IE
IP
INTERRUPT, SERIAL PORT AND TIMER BLOCKS
ALU
PC INCREMENTER
PSW
PROGRAM COUNTER
PSEN
ALE
EA
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
PORT 1 LATCH
PORT 6 LATCHES
PORT 3 LATCH
OSC
PORT 1 DRIVERS
PORT 6 DRIVERS
PORT 6 CONTROL/STATUS
PORT 3 DRIVERS
XTAL1
XTAL2
P1.0–P1.7
P6.0–P6.7
IDS
ODS
AFLAG
BFLAG
P3.0–P3.7
BD02660S

PIN CONFIGURATION

MNEMONIC	PIN NO. DIP	PIN NO. PLCC	TYPE	NAME AND FUNCTION
V_{SS}	50	54		**Ground:** 0V reference
V_{CC}	18	18		**Power Supply:** +5V
P0.0 – P0.7	17 – 10	17 – 10	I/O	**Port 0:** An 8-bit open drain, bidirectional I/O port. Port 0 is also the multiplexed low-order address and data bus during accesses to external memory, and outputs instruction bytes during program verification. External pull-ups are required during program verification. Port 0 can sink/source eight LS TTL inputs.
P1.0 – P1.7	23 – 30	27 – 34	I/O	**Port 1:** An 8-bit bidirectional I/O port with internal pull-ups. Port 1 receives the low-order address bytes during program verification. In the SC83C451, port 1 can sink/source three LS TTL inputs, and drive CMOS inputs without external pull-ups.
P2.0 – P2.7	2 – 9	2 – 9	I/O	**Port 2:** An 8-bit bidirectional I/O port with internal pull-ups. Port 2 emits the high-order address bytes during accesses to external memory and receives the high-order address bits and control signals during program verification in the SC83C451. Port 2 can sink/source three LS TTL inputs and drive CMOS inputs without external pull-ups.
P3.0 – P3.7	32 – 39	36 – 43	I/O	**Port 3:** An 8-bit bidirectional I/O port with internal pull-ups. Port 3 can sink/source three LS TTL inputs and drive CMOS inputs without external pull-ups. Port 3 also serves the SC83C451 special functions listed below:
	32	36	I	RXD (P3.0): Serial input port
	33	37	O	TXD (P3.1): Serial output port
	34	38	I	$\overline{\text{INT0}}$ (P3.2): External interrupt 0
	35	39	I	$\overline{\text{INT1}}$ (P3.3): External interrupt 1
	36	40	I	T0 (P3.4): Timer 0 external input
	37	41	I	T1 (P3.5): Timer 1 external input
	38	42	O	$\overline{\text{WR}}$ (P3.6): External data memory write strobe
	39	43	O	$\overline{\text{RD}}$ (P3.7): External data memory read strobe
P4.0 – P4.3	22 – 19		I/O	**Port 4:** A 4- or 8-bit bidirectional port with internal pull-ups. Port 4 can sink/source three LS TTL inputs and drive CMOS inputs without external pull-ups.
P4.0 – P4.7		26 – 19	I/O	
P5.0 – P5.7	40 – 47	44 – 51	I/O	**Port 5:** An 8-bit I/O port with internal pull-ups. Port 5 can sink/source three LS TTL inputs and drive CMOS inputs without external pull-ups.
P6.0 – P6.7	55 – 62	59 – 66	I/O	**Port 6:** A specialized 8-bit bidirectional I/O port with internal pull-ups. This special port can sink/source three LS TTL inputs and drive CMOS inputs without external pull-ups. Port 6 can be used in a strobed or non-strobed mode of operation, and in conjunction with four control pins that serve the functions listed below.
				Port 6 Control Lines:
$\overline{\text{ODS}}$	51	55	I	$\overline{\text{ODS}}$: Output data strobe
$\overline{\text{IDS}}$	52	56	I	$\overline{\text{IDS}}$: Input data strobe
BFLAG	53	57	I/O	BFLAG: A bidirectional I/O pin with internal pull-ups
AFLAG	54	58	I/O	AFLAG: A bidirectional I/O pin with internal pull-ups
RST	31	35	I	**RESET:** A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal pull-down resistor permits power-on reset using only a capacitor connected to the V_{CC}.
ALE	64	68	O	**Address Latch Enable:** An output for latching the low byte of the address during accesses to external memory. ALE is activated at a constant rate of ⅙ the oscillator frequency except during an external data memory access, at which time one ALE is skipped. ALE can sink/source eight LS TTL inputs and drive CMOS inputs without an external pull-up.
$\overline{\text{PSEN}}$	63	67	O	**Program Store Enable:** This output is the read strobe to external program memory. $\overline{\text{PSEN}}$ is activated twice each machine cycle during fetches from external program memory; however, when executing out of external program memory, two activations of $\overline{\text{PSEN}}$ are skipped during each access to external data memory. $\overline{\text{PSEN}}$ is not activated during fetches from internal program memory. $\overline{\text{PSEN}}$ can sink/source eight LS TTL inputs and drive CMOS inputs without an external pull-up.
$\overline{\text{EA}}$	1	1	I	**Instruction Execution Control:** When $\overline{\text{EA}}$ is held high, the CPU executes out of internal program memory, unless the program counter exceeds 0FFFH. When $\overline{\text{EA}}$ is held low, the CPU executes out of external program memory. $\overline{\text{EA}}$ must never be allowed to float.
XTAL1	49	53	I	**Crystal 1:** An input to the inverting amplifier that forms the oscillator. This input receives the external oscillator when an external oscillator is used.
XTAL2	48	52	O	**Crystal 2:** An output of the inverting amplifier that forms the oscillator. This pin should be floated when an external oscillator is used.

PORTS 4 and 5

Ports 4 and 5 are bidirectional I/O ports with internal pull-ups. Port 4 is an 8-bit port (PLCC version) or a 4-bit port (DIP Version). Port 4 and port 5 pins, with ones written to them, are pulled high by the internal pull-ups, and in that state can be used as inputs. Ports 4 and 5 are addressed at the special function register addresses shown in Table 1.

PORT 6

Port 6 is a special 8-bit bidirectional I/O port with internal pull-ups. (See Figure 1.) This port can be used as a standard I/O port, or in strobed modes of operation in conjunction with four special control lines: $\overline{\text{ODS}}$, $\overline{\text{IDS}}$, AFLAG, and BFLAG. Port 6 operating modes are controlled by the port 6 control status register (CSR). Port 6 and the CSR are addressed at the special function register addresses shown in Table 1. The following four control pins are used in conjunction with port 6:

$\overline{\text{ODS}}$: Output data strobe input for port 6. $\overline{\text{ODS}}$ can be programmed to control the port 6 output drivers and the output buffer full flag (OBF), or to clear only the OBF flag bit in the CSR (output-always mode). $\overline{\text{ODS}}$ is active low for output driver control. The OBF flag can be programmed to be cleared on the negative or positive edge of $\overline{\text{ODS}}$.

$\overline{\text{IDS}}$: Input data strobe input for port 6. $\overline{\text{IDS}}$ is used to control the port 6 input latch and input buffer full flag (IBF) bit in the CSR. The input data latch can be programmed to be transparent when the $\overline{\text{IDS}}$ is low and latched on the positive transition of $\overline{\text{IDS}}$, or to latch only on the positive transition of $\overline{\text{IDS}}$. Correspondingly, the IBF flag is set on the negative or positive transition of $\overline{\text{IDS}}$.

AFLAG: A bidirectional I/O pin. AFLAG can be programmed to be an output set high or low under program control, or to output the state of the output buffer full flag. AFLAG can also be programmed to be an input which selects whether the contents of the output buffer, or the contents of the port 6 control status register will be output on port 6. This feature grants complete port 6 status to external devices.

BFLAG: A bidirectional I/O pin. BFLAG can be programmed to be an output, set high or low under program control, or to output the state of the input buffer full flag. BFLAG can also be programmed to input an enable signal for port 6. When BFLAG is used as an enable input, port 6 output drivers are in the high impedance state, and the input latch does not respond to the $\overline{\text{IDS}}$ strobe when BFLAG is high. Both features are enabled when BFLAG is low. This feature facilitates the use of the SC83C451 in bused multiprocessor systems.

CONTROL STATUS REGISTER

The control status register (CSR) establishes the mode of operation for port 6 and indicates the current status of the port 6 I/O registers. All control status register bits can be read and written by the CPU, except bits 0 and 1, which are read only. Reset writes ones to bits 2 through 7, and writes zeros to bits 0 and 1.

CSR.0 — Input; Buffer Full Flag (IBF) (Read only)

The IBF bit is set to a logic 1 when port 6 data is loaded into the input buffer under control of $\overline{\text{IDS}}$. This can occur on the negative or positive edge of $\overline{\text{IDS}}$, as determined by CSR.2. IBF is cleared when the CPU reads the input buffer register.

CSR.1 — Output Buffer Full Flag (OBF) (Read only)

The OBF flag is set to a logic 1 when the CPU writes to the port 6 output data buffer. OBF is cleared by the positive or negative edge of $\overline{\text{ODS}}$, as determined by CSR.3.

CSR.2 — IDS Mode Select (IDSM)

When CSR.2 = 0, a low-to-high transition on the $\overline{\text{IDS}}$ pin sets the IBF flag. The port 6 input buffer is loaded on the $\overline{\text{IDS}}$ positive edge. When CSR.2 = 1, a high-to-low transition on the $\overline{\text{IDS}}$ pin sets the IBF flag. The port 6 input buffer is transparent when $\overline{\text{IDS}}$ is low, and latched when $\overline{\text{IDS}}$ is high.

CSR.3 — Output Buffer Full Flag Clear Mode (OBFC)

When CSR.3 = 1, the positive edge of the $\overline{\text{ODS}}$ input clears the OBF flag. When CSR.3 = 0, the negative edge of the $\overline{\text{ODS}}$ input clears the OBF flag.

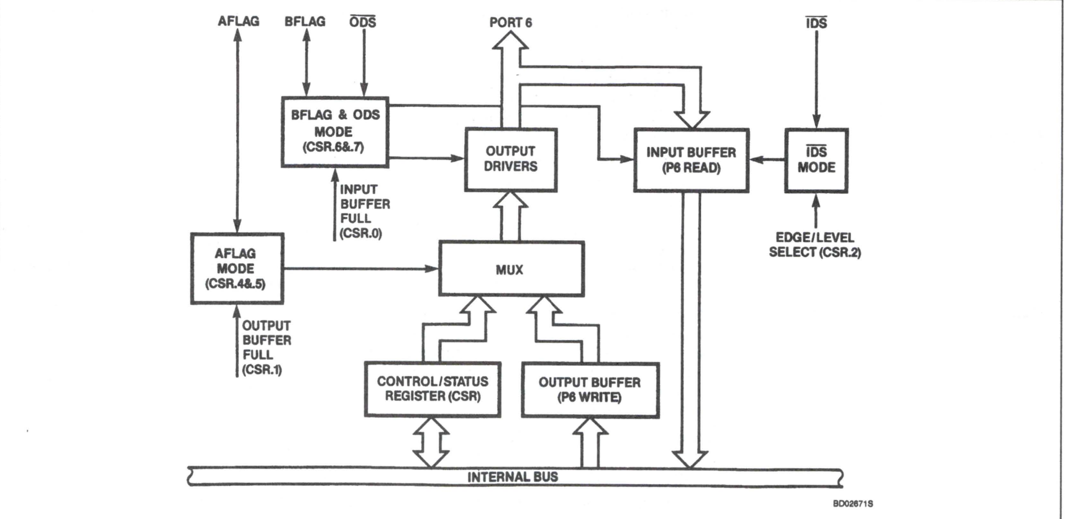


Figure 1. Port 6 Block Diagram

CSR.4, CSR.5 — AFLAG Mode Select (MA0, MA1)

Bits 4 and 5 select the mode of operation for the AFLAG pin, as follows:

MA1	MA0	AFLAG Function
0	0	Logic 0 output
0	1	Logic 1 output
1	0	OBF flag output (CSR.1)
1	1	Select (SEL) input mode

The select (SEL) input mode is used to determine whether the port 6 data register or the control status register is output on port 6. When the select feature is enabled, the AFLAG input controls the source of port 6 output data. A logic 0 on AFLAG input selects the port 6 data register, and a logic 1 on AFLAG input selects the control status register.

CSR.6, CSR.7 — BFLAG Mode Select (MB0, MB1)

Bits 6 and 7 select the mode of operation for the BFLAG pin, as follows:

MB1	MB0	BFLAG Function
0	0	Logic 0 output
0	1	Logic 1 output
1	0	IBF flag output (CSR.0)
1	1	Port enable ($\overline{PE}$)

In the port enable mode, $\overline{IDS}$ and $\overline{ODS}$ inputs are disabled when BFLAG input is high. When the BFLAG input is low, the port is enabled for I/O.

CONTROL STATUS REGISTER

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
(CSR)	MB1	MB0	MA1	MA0	OBFC	IDSM	OBF	IBF
	BFLAG mode select		AFLAG mode select		Output buffer flag clear mode	Input data strobe mode	Output buffer full flag	Input buffer full flag
	0/0 = Logic 0 output* 0/1 = Logic 1 output* 1/0 = IBF output 1/1 = $\overline{PE}$ input (0 = Select) (1 = disable I/O)		0/0 = Logic 0 output 0/1 = Logic 1 output 1/0 = OBF output* 1/1 = SEL input (0 = Data) (1 = Control/status)		0 = Negative edge of $\overline{ODS}$ 1 = Positive edge of $\overline{ODS}$	0 = Positive edge of $\overline{IDS}$ 1 = Low level of $\overline{IDS}$	0 = Output data buffer empty 1 = Output data buffer full	0 = Input data buffer empty 1 = Input data buffer full

*Output-always mode: MB1 = 0, MA1 = 1, and MA0 = 0. In this mode port 6 is always enabled for output. $\overline{ODS}$ only clears the OBF flag.

SPECIAL FUNCTION REGISTER ADDRESSES

Special function register addresses for the SC83C451 are identical to those of the SC80C51, except for the additional registers listed in Table 1.

Table 1. Special Function Register Addresses

REGISTER ADDRESS			BIT ADDRESS							
Name	Symbol	Address	MSB							LSB
Port 4	P4	C0	C7	C6	C5	C4	C3	C2	C1	C0
Port 5	P5	C8	CF	CE	CD	CC	CB	CA	C9	C8
Port 6 data	P6	D8	DF	DE	DD	DC	DB	DA	D9	D8
Port 6 control status	CSR	E8	EF	EE	ED	EC	EB	EA	E9	E8

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature range	0 to +70	°C
T_{STG}	Storage temperature range	-65 to +150	°C
	Voltage from V_{CC} to V_{SS} [2]	-0.5 to +6.5	V
	Voltage from any pin to V_{SS} [2]	-0.5 to V_{CC} + 0.5	V
PD	Power dissipation	200	mW

DC ELECTRICAL CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS Min	LIMITS Max	UNIT
V_{IL}	Input low voltage, except $\overline{EA}$		-0.5	0.2 V_{CC} -.1	V
V_{IL1}	Input low voltage, $\overline{EA}$		-0.5	0.2 V_{CC} -.3	V
V_{IH}	Input high voltage, except XTAL1, RST		0.2V_{CC} +.9	V_{CC} +0.5	V
V_{IH1}	Input high voltage, XTAL1, RST		0.7V_{CC}	V_{CC} +0.5	V
V_{OL}	Output low voltage, ports 1,2,3,4,5,6, AFLAG, BFLAG	I_{OL} = 1.6mA[4]		0.45	V
V_{OL1}	Output low voltage, port 0, ALE, $\overline{PSEN}$	I_{OL} = 3.2mA[4]		0.45	V
V_{OH}	Output high voltage, ports 1,2,3,4,5,6, AFLAG, BFLAG	I_{OH} = -60μA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25μA	0.75V_{CC}		V
		I_{OH} = -10μA	0.9V_{CC}		V
V_{OH1}	Output high voltage, port 0 in external bus mode, ALE, $\overline{PSEN}$	I_{OH} = -400μA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -150μA	0.75V_{CC}		V
		I_{OH} = -40μA[5]	0.9V_{CC}		V
I_{IL}	Logical 0 input current, ports 1,2,3,4,5,6, AFLAG, BFLAG	V_{IN} = 0.45V		-50	μA
I_{TL}	Logical 1 to 0 transition current, ports 1,2,3,4,5,6, AFLAG, BFLAG	V_{IN} = 2V		-650	μA
I_{LI}	Input leakage, current port 0, $\overline{EA}$, $\overline{IDS}$, $\overline{ODS}$	0.45 < V_{IN} < V_{CC}		+10	μA
R_{RST}	Reset pulldown resistor		50	150	kΩ
C_{IO}	Pin capacitance	Test freq = 1MHz, T_A = 25°C		10	pF
I_{PD}	Power-down current	V_{CC} = 2 to 6V[6]		50	μA

MAXIMUM I_{CC} (mA)

	OPERATING[7]			IDLE[8]		
Freq. V_{CC}	4V	5V	6V	4V	5V	6V
0.5MHz 3.5MHz 8.0MHz 12MHz 16MHz	To be determined	To be determined	To be determined	To be determined	To be determined	To be determined

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 20\%$, $V_{SS} = 0V$, load capcitance for port 0, ALE, and $\overline{PSEN} = 100pF$, load capacitance for all other outputs = 80 pF[3]

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK (f = $1/t_{CLCL}$)		
		Min	Max	Min	Max	Unit
$1/t_{CLCL}$	Speed range (see ordering code)					MHz
	0			3.5	12	
	1			3.5	16	
	2			0.5	12	
t_{LHLL}	ALE pulse width	127		$2t_{CLCL} - 40$		ns
t_{AVLL}	Address valid to ALE low	28		$t_{CLCL} - 55$		ns
t_{LLAX}	Address hold after ALE low	48		$t_{CLCL} - 35$		ns
t_{LLIV}	ALE low to valid instr in		234		$4t_{CLCL} - 100$	ns
t_{LLPL}	ALE low to $\overline{PSEN}$ low	43		$t_{CLCL} - 40$		ns
t_{PLPH}	$\overline{PSEN}$ pulse width	205		$3t_{CLCL} - 45$		ns
t_{PLIV}	$\overline{PSEN}$ low to valid instr in		145		$3t_{CLCL} - 105$	ns
t_{PXIX}	Input instr hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input instr float after $\overline{PSEN}$		59		$t_{CLCL} - 25$	ns
t_{AVIV}	Address to valid instr in		312		$5t_{CLCL} - 105$	ns
t_{PLAZ}	$\overline{PSEN}$ low to address float		10		10	ns
t_{RLRH}	$\overline{RD}$ pulse width	400		$6t_{CLCL} - 100$		ns
t_{WLWH}	$\overline{WR}$ pulse width	400		$6t_{CLCL} - 100$		ns
t_{RLDV}	$\overline{RD}$ low to valid data in		252		$5t_{CLCL} - 165$	ns
t_{RHDX}	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data float after $\overline{RD}$		97		$2t_{CLCL} - 70$	ns
t_{LLDV}	ALE low to valid data in		517		$8t_{CLCL} - 150$	ns
t_{AVDV}	Address to valid data in		585		$9t_{CLCL} - 165$	ns
t_{LLWL}	ALE low to $\overline{RD}$ or $\overline{WR}$ low	200	300	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	23		$t_{CLCL} - 60$		ns
t_{WHQX}	Data hold after $\overline{WR}$	33		$t_{CLCL} - 50$		ns
t_{RLAZ}	$\overline{RD}$ low to address float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ high to ALE high	43	123	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
Port 6 input (input rise and fall times 5ns)						
t_{ILIH}	$\overline{IDS}$ width	270		$3t_{CLCL} + 20$		ns
t_{DVIH}	Data setup to $\overline{IDS}$ high	0		0		ns
t_{IHDX}	Data hold after $\overline{IDS}$	30		30		ns
t_{FLIL}	$\overline{PE}$ to $\overline{IDS}$	25		25		ns
t_{IVFV}	$\overline{IDS}$ to BFLAG (IBF) delay		130		130	ns
Port 6 output						
t_{OLOH}	$\overline{ODS}$ width	270		$3t_{CLCL} + 20$		ns
t_{FVDV}	SEL to data out delay		85		85	ns
t_{OLDV}	$\overline{ODS}$ to data out delay		80		80	ns
t_{OHDZ}	$\overline{ODS}$ to data float delay		35		35	ns
t_{OVFV}	$\overline{ODS}$ to AFLAG (OBF) delay		100		100	ns
t_{FLDV}	$\overline{PE}$ to data out delay		120		120	ns
t_{OHFH}	$\overline{ODS}$ high to AFLAG (SEL) delay	100		100		ns

NOTES:

1. Stress above those listed under Absolute Maximum Rating may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection on its internal devices from damaging effects of excessive ststic charge. Nonetheless it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maximum.
3. Parameters are valid over operating temperature range unless otherwise specified.
4. Capacitive loading on ports 0 and 2 can cause spurious noise to be superimposed on the $V^{OLS}$of ALE and ports 1 and 3. The noise is caused by external bus capacitance discharging into the port 0 and pot 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases it could be desirable to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger STROBE input.
5. Capacitive loading on ports 0 and 2 can cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall before the 0.9 V_{CC} specification when the address bits are stabilizing.
6. Power-down I_{CC} is measured with all output pins disconnected; $\overline{EA}$ = port 0 = V_{CC}; XTAL2 N.C.; RST = V_{SS}.
7. I_{CC} is measured with all output pins disconnected; XTAL1 driven with t_{CLCH}, t_{CHCL} = 5ns, $V_{IL} = V_{SS} + 0.5V$, $V_{IH} = V_{CC} - 0.5V$; XTAL1 N.C.; $\overline{EA}$ = RST = Port 0 = V_{CC}. I_{CC} will be slightly higher if a crystal oscillator is used.
8. Idle I_{CC} is measured with all output pins disconnected; XTAL1 driven with t_{CLCH}, t_{CHCL} = 5ns, $V_{IL} = V_{SS} + 0.5V$, $V_{IH} = V_{CC} - 0.5V$; XTAL N.C.; port 0 = V_{CC}; $\overline{EA}$ = RST = V_{SS}.

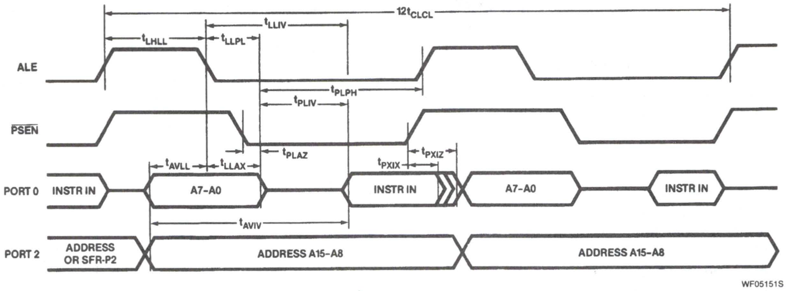

12tCLCL
tLLIV
tLHLL
tLLPL
ALE
tPLPH
tPLIV
PSEN
tPLAZ
tPXIZ
tAVLL
tLLAX
tPXIX
PORT 0
INSTR IN
A7-A0
INSTR IN
A7-A0
INSTR IN
tAVIV
PORT 2
ADDRESS OR SFR-P2
ADDRESS A15-A8
ADDRESS A15-A8
WF05151S

Figure 2. Program Memory Read Cycle

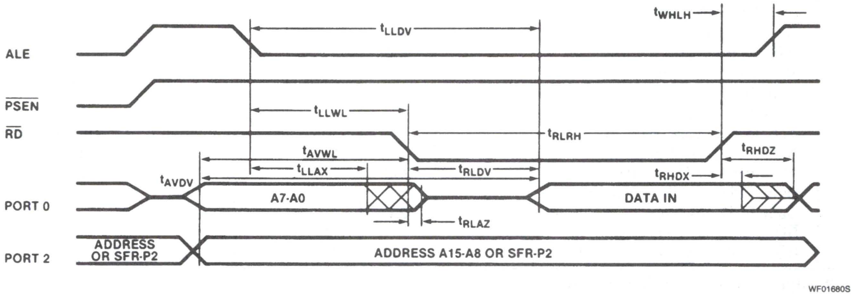

tWHLH
tLLDV
ALE
PSEN
tLLWL
RD
tRLRH
tAVWL
tRHDZ
tLLAX
tRLDV
tRHDX
tAVDV
PORT 0
A7-A0
DATA IN
tRLAZ
PORT 2
ADDRESS OR SFR-P2
ADDRESS A15-A8 OR SFR-P2
WF01680S

Figure 3. Data Memory Read Cycle

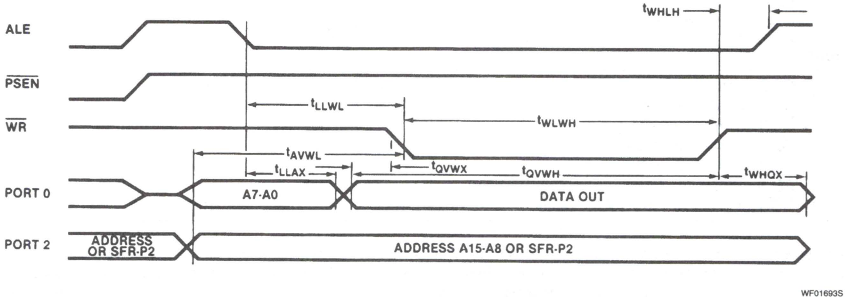

tWHLH
ALE
PSEN
tLLWL
WR
tWLWH
tAVWL
tLLAX
tQVWX
tQVWH
tWHQX
PORT 0
A7-A0
DATA OUT
PORT 2
ADDRESS OR SFR-P2
ADDRESS A15-A8 OR SFR-P2
WF01693S

Figure 4. Data Memory Write Cycle

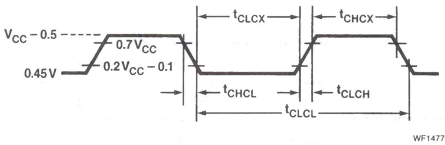

t_{CLCX}
t_{CHCX}
$V_{CC}-0.5$
$0.7V_{CC}$
$0.45V$
$0.2V_{CC}-0.1$
t_{CHCL}
t_{CLCH}
t_{CLCL}
WF14770S

Figure 5. External Clock Drive Waveform

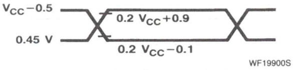

$V_{CC}-0.5$
$0.45\ V$
$0.2\ V_{CC}+0.9$
$0.2\ V_{CC}-0.1$
WF19900S

Figure 6. AC Testing Input, Output Waveforms[1]

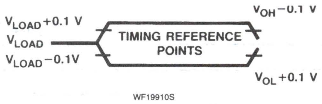

$V_{OH}-0.1\ V$
$V_{LOAD}+0.1\ V$
V_{LOAD}
$V_{LOAD}-0.1V$
TIMING REFERENCE POINTS
$V_{OL}+0.1\ V$
WF19910S

Figure 7. Float Waveforms[2]

NOTES:

1. AC inputs during testing are driven at $V_{CC}-0.5$ for a logic "1" and 0.45V for a logic "0". Timing measurements are made at V_{IH} min. for a logic "1" and V_{IL} max. for a logic "0".
2. For timing purposes a port pin is no longer floating when a 100mV change from load voltage occurs, and begins to flot when a 100mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} > \pm 20mA$.

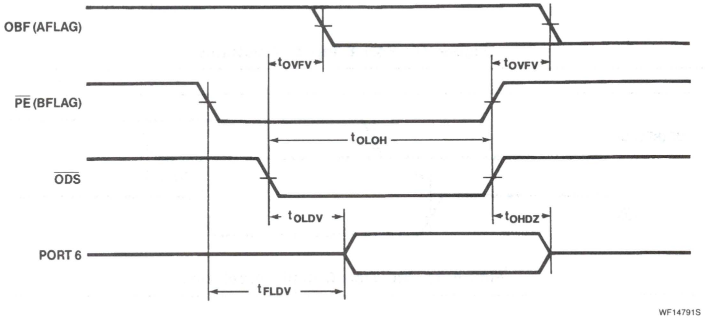

OBF (AFLAG)
t_{OVFV}
t_{OVFV}
PE (BFLAG)
t_{OLOH}
ODS
t_{OLDV}
t_{OHDZ}
PORT 6
t_{FLDV}
WF14791S

Figure 8. Port 6 Output Waveforms

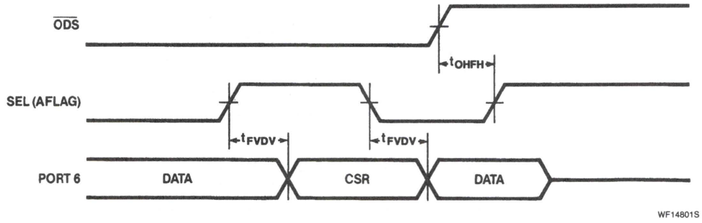

ODS
tOHFH
SEL (AFLAG)
tFVDV
tFVDV
PORT 6
DATA
CSR
DATA
WF14801S

Figure 9. Port 6 Select Mode Waveforms

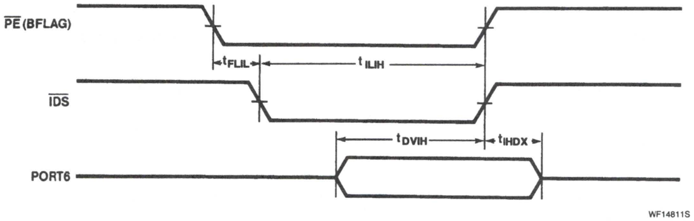

PE (BFLAG)
tFLIL
tILIH
IDS
tDVIH
tIHDX
PORT6
WF14811S

Figure 10. Port 6 Input Waveforms

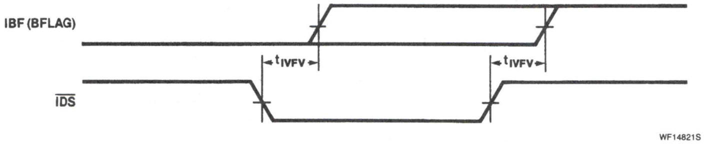
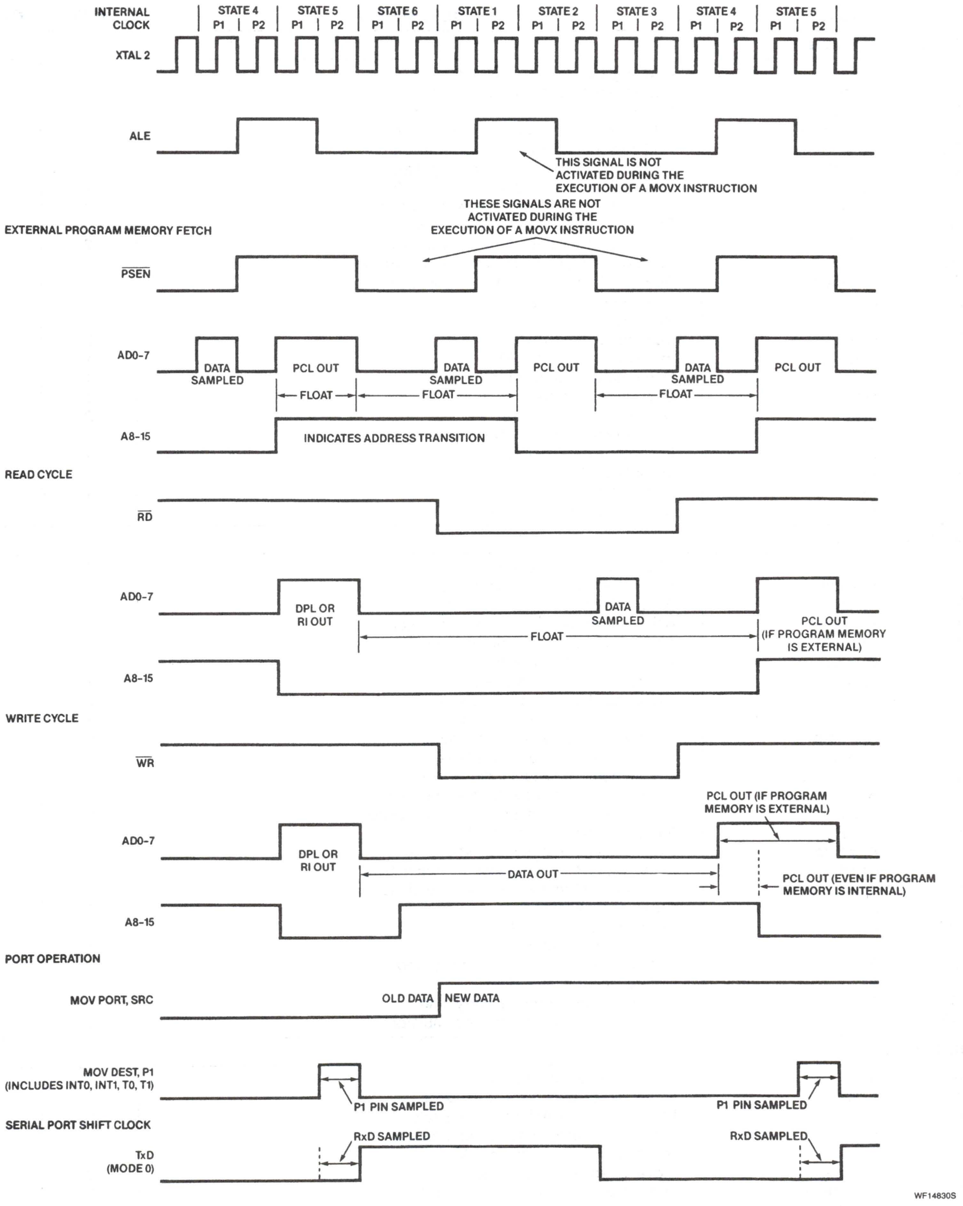

IBF (BFLAG)
tIVFV
tIVFV
IDS
WF14821S

Figure 11. IBF Flag Output Waveforms


INTERNAL CLOCK
STATE 4 P1 P2
STATE 5 P1 P2
STATE 6 P1 P2
STATE 1 P1 P2
STATE 2 P1 P2
STATE 3 P1 P2
STATE 4 P1 P2
STATE 5 P1 P2
XTAL 2
ALE
THIS SIGNAL IS NOT ACTIVATED DURING THE EXECUTION OF A MOVX INSTRUCTION
THESE SIGNALS ARE NOT ACTIVATED DURING THE EXECUTION OF A MOVX INSTRUCTION
EXTERNAL PROGRAM MEMORY FETCH
PSEN
AD0-7
DATA SAMPLED
PCL OUT
DATA SAMPLED
PCL OUT
DATA SAMPLED
PCL OUT
FLOAT
FLOAT
FLOAT
A8-15
INDICATES ADDRESS TRANSITION
READ CYCLE
RD
AD0-7
DPL OR RI OUT
DATA SAMPLED
FLOAT
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL)
A8-15
WRITE CYCLE
WR
PCL OUT (IF PROGRAM MEMORY IS EXTERNAL)
AD0-7
DPL OR RI OUT
DATA OUT
PCL OUT (EVEN IF PROGRAM MEMORY IS INTERNAL)
A8-15
PORT OPERATION
MOV PORT, SRC
OLD DATA
NEW DATA
MOV DEST, P1 (INCLUDES INT0, INT1, T0, T1)
P1 PIN SAMPLED
P1 PIN SAMPLED
SERIAL PORT SHIFT CLOCK
RxD SAMPLED
RxD SAMPLED
TxD (MODE 0)
WF14830S

NOTE:
This diagram indicates when signals are clocked internally, however, the time it takes the signals to propagate to the pins ranges from 25ns to 125ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component to component. Typically (T_A = 25°C, fully loaded), $\overline{RD}$ and $\overline{WR}$ propagation delays are approximately 50ns; propagation delays for other signals are approximately 85ns. These timing delays are incorporated in the AC specifications.

Figure 12. Timing Waveforms

I/O Expanded CMOS Microcontroller SC80C451/SC83C451

Table 2. Instruction Set

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Arithmetic operations				
ADD	A,Rn	Add register to accumulator	1	1
ADD	A,direct	Add direct byte to accumulator	2	1
ADD	A,@Ri	Add indirect RAM to accumulator	1	1
ADD	A,R#data	Add immediate data to accumulator	2	1
ADDC	A,Rn	Add register to accumulator with carry	1	1
ADDC	A,direct	Add direct byte to A with carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1
ADDC	A,#data	Add immediate data to A with carry flag	2	1
SUBB	A,Rn	Subtract register from A with borrow	1	1
SUBB	A,direct	Subtract direct byte from A with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A w/borrow	1	1
SUBB	A,#data	Subtract immed. data from A w/borrow	2	1
INC	A	Increment accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment data pointer	1	2
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal adjust accumulator	1	1
Logical operations				
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate A left through the carry flag	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate A right through carry flag	1	1
SWAP	A	Swap nibbles within the accumulator	1	1
Data transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2

Table 2. Instruction Set (Continued)

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Data transfer (continued)				
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to A	1	2
MOVC	A,@A + PC	Move code byte relative to PC to A	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to external RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order digit ind. RAM w/A	1	1
Boolean variable manipulation				
CLR	C	Clear carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry flag	2	2
ANL	C,/bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry flag	2	2
ORL	C,/bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry flag	2	1
MOV	bit,C	Move carry flag to direct bit	2	2
Program and machine control				
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long jump	3	2
SJMP	rel	Short jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if accumulator is zero	2	2
JNZ	rel	Jump if accumulator is not zero	2	2
JC	rel	Jump if carry flag is set	2	2
JNC	rel	Jump if no carry flag	2	2
JB	bit,rel	Jump if direct bit set	3	2
JNB	bit,rel	Jump if direct bit not set	3	2
JBC	bit,rel	Jump if direct bit is set & clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & jump if not equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & jump if not equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg. & jump if not equal	3	2
CJNE	@Ri,#data,rel	Comp. immed. to ind. & jump if not equal	3	2
DJNZ	Rn,rel	Decrement register & jump if not zero	2	2
DJNZ	direct,rel	Decrement direct & jump if not zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn – Working register R0-R7
direct – 128 internal RAM locations, any I/O port, control or status register
@Ri – Indirect Internal RAM location addressed by register R0 or R1
#data – 8-bit constant included in instruction
#data16 – 16-bit constant included as bytes 2 & 3 of instruction
bit – 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr16 – Destination address for LCALL & LJMP may be anywhere within the 64-kilobyte program memory address space.
addr11 – Destination address for ACALL & AJMP will be within the same 2-kilobyte page of program memory as the first byte of the following instruction.
rel – SJMP and all conditional jumps include an 8-bit offset byte. Range is +127 – 128 bytes relative to first byte of the following instruction.

Table 3. Instruction Opcodes in Hexadecimal Order

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,#data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr,A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data

Table 3. Instruction Opcodes in Hexadecimal Order (Continued)

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A + PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A + DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Signetics

SC87C451
CMOS Single-Chip 8-Bit EPROM Microcontroller

Objective Specification (Brief)

Microprocessor Products

DESCRIPTION

The Signetics SC87C451 is an I/O expanded, single-chip microcontroller fabricated with Signetics high-density CMOS technology. Signetics epitaxial substrate minimizes latch-up sensitivity.

The SC87C451 has 4K of EPROM on chip as program memory and is otherwise identical to the SC83C451.

The SC87C451 is a functional extension of the SC80C31/SC80C51 microcontroller with three additional I/O ports and four I/O control lines. The PLCC version has a total of 68 pins. Four control lines associated with port 6 facilitate high speed asynchronous I/O functions.

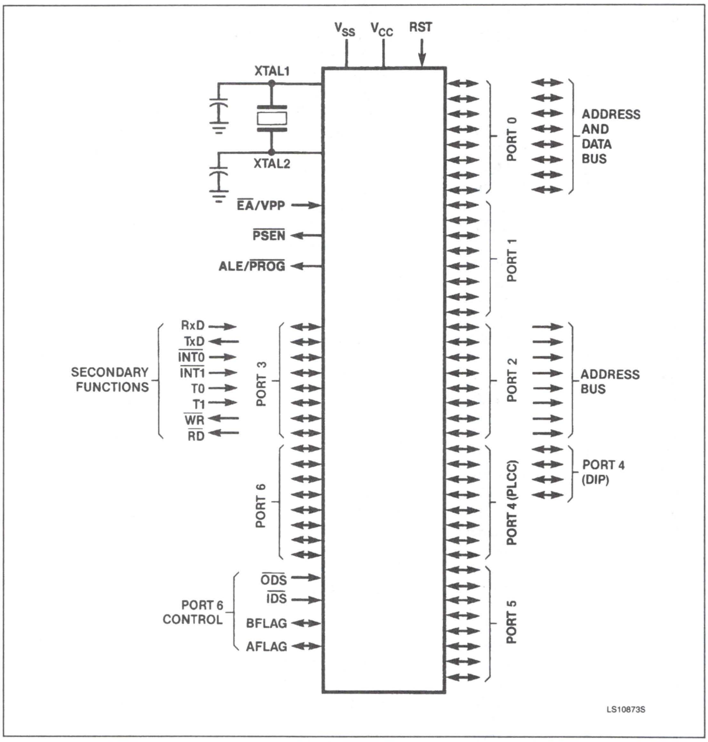
The SC87C451 includes a 4K × 8 EPROM; a 128 × 8 RAM; 68 (PLCC) or 64 (DIP) I/O lines; two 16-bit timer/counters; a five source, two priority level, nested interrupt structure; a serial I/O port for either a full duplex UART, I/O expansion, or multiprocessor communications; and on-chip oscillator and clock circuits.

The SC87C451 has two software selectable modes of reduced activity for further power reduction: idle mode and power-down mode. Idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. Power-down mode freezes the oscillator, causing all other chip functions to be inoperative while maintaining the RAM contents.

FEATURES

- **User programmable microcontroller**
- **SC80C51 based architecture**
- **68-pin PLCC and 64-pin DIP packages:**
  - **Seven 8-bit I/O ports (PLCC version)**
  - **Six 8-bit ports and one 4-bit port (DIP version)**
- **Port 6 features:**
  - **8 data pins**
  - **4 control pins**
  - **Direct MPU bus interface**
  - **Parallel printer interface**
- **On the microcontroller**
  - **4K × 8 EPROM**
  - **128 × 8 RAM**
  - **Two 16-bit counter/timers**
  - **Two external interrupts**
- **External memory addressing capability**
  - **64K ROM and 64K RAM**
- **Low power consumption:**
  - **Normal operation: less than 24mA at 5V, 12Mhz**
  - **Idle mode**
  - **Power-down mode**

LOGIC SYMBOL


V_{SS}
V_{CC}
RST
XTAL1
XTAL2
EA/VPP
PSEN
ALE/PROG
SECONDARY FUNCTIONS
RxD
TxD
INT0
INT1
T0
T1
WR
RD
PORT 3
PORT 6
PORT 6 CONTROL
ODS
IDS
BFLAG
AFLAG
PORT 0
PORT 1
PORT 2
PORT 4 (PLCC)
PORT 5
ADDRESS AND DATA BUS
ADDRESS BUS
PORT 4 (DIP)
LS10873S

Signetics

SC83C751
CMOS Single-Chip 8-Bit Microcontroller

Objective Specification

Microprocessor Products

DESCRIPTION

The SC83C751 offers many of the advantages of the SC80C51 architecture in a small package and at low cost.

The Signetics SC83C751 microcontroller is fabricated with Signetics high-density CMOS technology. Signetics epitaxial substrate minimizes CMOS latch-up sensitivity.

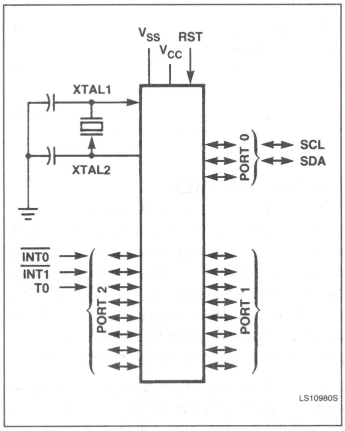
The SC83C751 contains a 2K × 8 ROM; a 64 × 8 RAM; 19 I/O lines; a 16-bit counter/timer; a five-source, fixed-priority interrupt structure; a bidirectional Inter-Integrated Circuit (I^2C) serial bus interface; and an on chip oscillator.

The onboard Inter-Integrated Circuit (I^2C) bus interface allows the SC83C751 to operate as a master or slave device on the I^2C small area network. This capability facilitates I/O and RAM expansion; access to EEPROM; processor-to-processor communication; and efficient interface to a wide variety of dedicated I^2C peripherals.

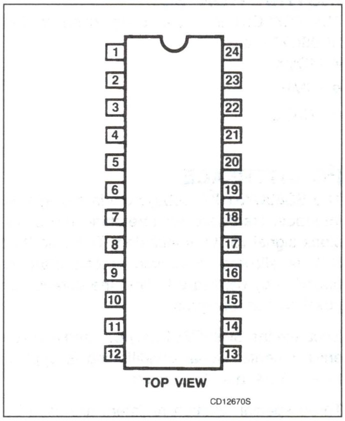
FEATURES

- **SC80C51 based architecture**
- **Inter-Integrated Circuit (I^2C) serial bus interface**
- **Small package sizes**
  - **24 pin DIP**
  - **24 pin SOL**
- **One-time programmable EPROM version SC87C751**
- **Wide power supply voltage range**
- **Wide oscillator frequency range**
- **Low power consumption:**
  - **Normal operation: less than 24mA at 5V, 12Mhz**
  - **Idle mode**
  - **Power-down mode**
- **2K × 8 ROM, 64 × 8 RAM**
- **16-bit counter/timer**
- **Boolean processor**
- **CMOS and TTL compatible**

LOGIC SYMBOL



PIN CONFIGURATION



ARCHITECTURE

Features of the SC83C751 are identical to the SC80C51, except as noted. Every attempt has been made to preserve program compatibility.

INSTRUCTION SET

The SC80C51 instructions not used on the SC83C751 are:

- MOVX
- LJMP
- LCALL

I^{2}C INTERFACE

The SC83C751 I^{2}C subsystem is a single-bit interface. Hardware for stretching the serial clock signal (SCL) is included. The time that SCL is allowed to remain in one state is bounded by hardware to limit the duration of possible bus hangups.

Because the SC83C751 may be used in noisy environments, signal conditioning is applied to the SDA and SCL inputs.

Three special function registers are used by the interface. They are:

- ICON — I^{2}C control register
- IDAT — I^{2}C data register
- ICFG — I^{2}C configuration register

PORT 0

Port 0 is a 3-bit open-drain bidirectional port. When port 0 is read or written, the data occupies the low-order bits of the 8-bit location. The alternate functions of Port 0 also serve the serial I^{2}C interface. To comply with the I^{2}C specification, P0.0 and P0.1 are open-drain bidirectional I/O pins with the following electrical characteristics:

Inputs:

VIL = MAX 0.3 × VCC
VIH = MIN 0.7 × VCC
VTH-VIL (Hysteresis) = 300mV MIN

Drivers:

VOL = 0.4V MAX @ IOL = 3mA

Driver, Receiver Combined:

IL = −10mA MAX @ V = 0.4
IH = 10mA MAX @ V = 0.9 × VCC
C = 10 pF MAX

IDLE MODE

The circuits that remain operative in idle mode are:

- Counter/Timer 0
- I^{2}C Interface
- All interrupt circuitry

INTERRUPTS

There are five interrupt sources of fixed priority. In order of descending priority, they are:

- Serial I^{2}C bus
- $\overline{\text{INT0}}$
- Counter/Timer
- $\overline{\text{INT1}}$
- Timer 1

Interrupts are controlled by the interrupt enable register:

IE Register

7	6	5	4	3	2	1	0
EA	—	—	EIIC	—	EX1	ET	EX0

DF06700S

EA	- Enable all interrupts. (Cleared on reset)
EIIC	- Enable I^{2}C interrupt
EX1	- Enable external interrupt 1
ET	- Enable timer interrupt
EX0	- Enable external interrupt 0

Interrupt Vectors

EVENT	MEMORY ADDRESS
Reset	000
External $\overline{\text{INT0}}$	003
Counter/Timer 0	00B
External $\overline{\text{INT1}}$	013
Timer I	01B
I$_2$C Serial	023

COUNTER/TIMER

The SC83C751 has one 16-bit counter/timer. In timer mode the timer register is incremented at 1/12 the oscillator frequency. In counter mode the register is incremented once for every 1-to-0 transition on the external input pin. Maximum input frequency is 1/24 the oscillator frequency because the external input is sampled once every machine cycle.

Counter/timer operation can be gated by an external source via the $\overline{\text{INT0}}$ pin. This feature is enabled by the gate bit in TCON.

The counter/timer automatically reloads on carry out of the timer register. The auto-reload registers and the timer registers can be read or written while the counter/timer is operating.

On carry out of the timer register, the timer flag bit is set. The counter/timer interrupt vector is executed if the counter/timer interrupt is enabled.

The counter/timer is controlled by the TCON register:

TCON Register

7	6	5	4	3	2	1	0
GATE	C/T	TF	TR	IE0	IT0	IE1	IT1

DF06710S

GATE — When this bit is set, the counter/timer can be controlled by the $\overline{\text{INT0}}$ pin. The counter/timer is enabled when the $\overline{\text{INT0}}$ pin is high and the counter/timer enable bit (TR) is set. The counter/timer is disabled when the INT0 pin is low or the TR bit is reset.

C/T — When this bit is set, the counter/timer operates as a counter. Negative logic transitions of the T1 pin are counted. When this bit is set to zero, the counter/timer operates as a timer and is incremented by the internal clock.

TF — The timer flag bit is set by hardware on the overflow of the 16-bit counter/timer; it is cleared when the interrupt vector is executed by the processor.

TR — When this bit is set, the counter/timer is enabled for operation.

IE0 — The interrupt edge flag is set by hardware when an external interrupt edge is detected on $\overline{\text{INT0}}$, and is cleared when the interrupt is serviced.

IT0 — When this bit is set, $\overline{\text{INT0}}$ is negative edge triggered. When this bit is cleared, $\overline{\text{INT0}}$ is low-level sensitive.

IE1 — The interrupt edge flag is set by hardware when an external interrupt edge is detected on $\overline{\text{INT1}}$, and is cleared when the interrupt is serviced.

IT1 — When this bit is set, $\overline{\text{INT1}}$ is negative edge triggered. When this bit is cleared, $\overline{\text{INT1}}$ is low-level sensitive.

I^{2}C LICENSE

Purchase of Signetics or Philips I^{2}C components conveys a license under the Philips I^{2}C patent rights to use these components in an I^{2}C system, provided that the system conforms to the I^{2}C standard specification as defined by Philips.

Signetics

Microprocessor Products

SC80C552/SC83C552

CMOS Automotive and Industrial Microcontroller

Objective Specification

DESCRIPTION

The Signetics SC80C552/SC83C552 is an SC80C31/SC80C51 expanded by the addition of a 10-bit A/D converter; eight analog inputs; eight timed outputs; a watchdog timer; two pulse with modulation outputs; and a serial interface supporting the I^2C protocol. The on-chip ROM has been increased to 8K bytes, and the RAM increased to 256 bytes. The original peripherals on the SC80C31/SC80C51 have been retained, including two timers, the UART, the power-down mode, and the idle mode.

The SC80C552/SC83C552 is packaged in a 68-pin PLCC package. This 68-pin PLCC package provides the required extra I/O pins and permits surface mounting, but occupies less space than a 40-pin DIL package.

The 10-bit A/D converter receives input via an on-chip, 8-channel analog multiplexer. Conversion can be initiated by software or by an external input. The two pulse width modulated outputs can be used as analog outputs, because the pulse width represents the analog level.

Timer 2 is a 16-bit timer that features four 16-bit capture inputs, two timed toggle outputs, and six set/reset outputs. Timer 2 can be used in many repetitive timing applications, such as a fuel injection engine.

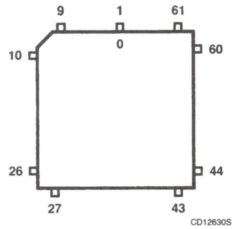
Timer 3 (the watchdog timer) has an added load enable flag for extra protection against runaway programs.

In addition to the UART serial interface inherited from the SC80C31/SC80C51, the SC80C552/SC83C552 has an I^2C serial interface. The I^2C is a multimaster protocol that permits several microcontrollers to communicate with each other, share RAM or EEPROM memories, or to communicate with peripherals over a two-wire bus.

FEATURES

- **SC80C51 CPU and instruction set**
- **8K bytes ROM (SC83C552 only)**
- **256 bytes RAM**
- **Timers 0 and 1**
  - **(Same as the SC80C31/SC80C51)**
- **Timer 2**
  - **Four 16-bit capture inputs**
  - **Six set/reset outputs**
  - **Two toggle outputs**
- **Timer 3**
  - **Watchdog timer with enable latch**
- **10-bit A/D Converter**
  - **8 Multiplexed inputs**
- **Two 8-bit analog outputs**
  - **Based on pulse width modulation**
- **Six 8-bit I/O ports**
- **I^2C serial I/O port**
  - **Multimaster networks**
- **UART**
  - **Full duplex**
- **Power saving**
  - **Idle mode**
  - **Power-down mode**
- **External memory expansion**

PIN CONFIGURATION

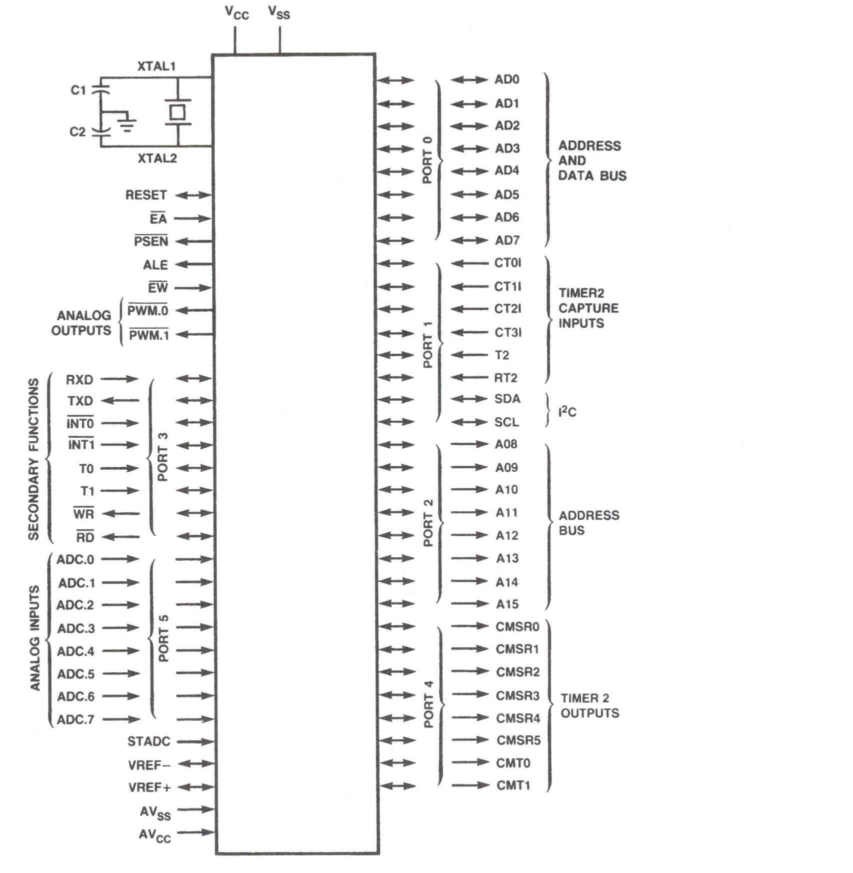


Pin	Function	Pin	Function
1	P5.0/ADC.0	35	XTAL2
2	V_{CC}	36	XTAL1
3	STADC	37	V_{SS}
4	PWM0N	38	V_{SS}
5	PWM1N	39	P2.0/A8
6	$\overline{EW}$	40	P2.1/A9
7	P4.0/CMSR0	41	P2.2/A10
8	P4.1/CMSR1	42	P2.3/A11
9	P4.2/CMSR2	43	P2.4/A12
10	P4.3/CMSR3	44	P2.5/A13
11	P4.4/CMSR4	45	P2.6/A14
12	P4.5/CMSR5	46	P2.7/A15
13	P4.6/CMT0	47	$\overline{PSEN}$
14	P4.7/CMT1	48	ALE
15	RST	49	$\overline{EA}$
16	P1.0/CT0I	50	P0.7/AD.7
17	P1.1/CT1I	51	P0.6/AD.6
18	P1.2/CT2I	52	P0.5/AD.5
19	P1.3/CT3I	53	P0.4/AD.4
20	P1.4/T2	54	P0.3/AD.3
21	P1.5/RT2	55	P0.2/AD.2
22	P1.6/SCL	56	P0.1/AD.1
23	P1.7/SDA	57	P0.0/AD.0
24	P3.0/RXD	58	AV_{ref+}
25	P3.1/TXD	59	AV_{ref-}
26	P3.2/$\overline{INT0}$	60	AV_{SS}
27	P3.3/$\overline{INT1}$	61	AV_{CC}
28	P3.4/T0	62	P5.7/ADC.7
29	P3.5/T1	63	P5.6/ADC.6
30	P3.6/$\overline{WR}$	64	P5.5/ADC.5
31	P3.7/$\overline{RD}$	65	P5.4/ADC.4
32	—	66	P5.3/ADC.3
33	—	67	P5.2/ADC.2
34	—	68	P5.1/ADC.1

PIN DESCRIPTION

MNEMONIC	PIN NO PLCC	TYPE	NAME & FUNCTION
V_{SS}	37, 38		Ground
V_{CC}	2		Positive power supply
AV_{SS}	60		Analog ground
AV_{CC}	61		Positive power supply for analog part
AV_{ref+}	59		High end of ADC reference resistor
AV_{ref-}	58		Low end of ADC reference resistor
XTAL1	36	I	**Crystal 1:** Input for clock source or timing device/crystal
XTAL2	35	O	**Crystal 2:** Output for timing device/crystal
RST	15	I/O	**Reset:** Input to reset the device. Outputs a pulse on T3 overflow to reset other devices.
$\overline{\text{EW}}$	6	I	**Enable Watchdog:** Input to enable the watchdog timer, and to disable the power-down mode.
$\overline{\text{PSEN}}$	47	O	**Program Store Enable:** Output for external ROM access
ALE	48	O	**Address Latch Enable:** Strobe output to latch A0 - A7
$\overline{\text{EA}}$	49	I	**External Access:** Input to enable external ROM access
STADC	3	I	**Start ADC:** Input to start A/D conversion.
$\overline{\text{PWMO}}$	4	O	**PWM.0:** Output, active low
$\overline{\text{PWM1}}$	5	O	**PWM.1:** Output, active low
P0 - P0.7	57 - 50	I/O	**Port 0:** 8 - bit bidirectional I/O port. Alternate function: Multiplexed low byte of external memory address and data byte.
P1 - P1 - 7	16 - 23	I/O	**Port 1:** 8 - bit Quasi-bidirectional I/O port. Alternate function: Timer 2 inputs.
	22		I^2C Serial clock line, open drain driver.
	23		I^2C Serial data line, open drain driver.
P2 - P2.7	39 - 46	I/O	**Port 2:** 8 - bit quasi-bidirectional I/O port. Alternate function: High byte of external memory address.
P3 - P3.7	24 - 31	I/O	**Port 3:** 8 - bit quasi-bidirectional I/O port. Alternate function: SIO 0 I/O pins external interrupt inputs, timer inputs, read and write synchronous outputs for external memory access.
P4 - P4.7	7 - 14	I/O	**Port 4:** 8 - bit quasi-bidirectional I/O port. Alternate function: Timer 2 outputs.
P5 - P5.7	1, 68 - 62	I	**Port 5:** 8 - bit input-only port. Alternate function: Inputs to A/D converter.

LOGIC SYMBOL


V_{CC} V_{SS}
XTAL1
C1
C2
XTAL2
RESET
EA
PSEN
ALE
EW
ANALOG OUTPUTS
PWM.0
PWM.1
SECONDARY FUNCTIONS
RXD
TXD
INT0
INT1
T0
T1
WR
RD
PORT 3
ANALOG INPUTS
ADC.0
ADC.1
ADC.2
ADC.3
ADC.4
ADC.5
ADC.6
ADC.7
PORT 5
STADC
VREF–
VREF+
AVSS
AVCC
PORT 0
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
ADDRESS AND DATA BUS
PORT 1
CT0I
CT1I
CT2I
CT3I
T2
RT2
TIMER2 CAPTURE INPUTS
SDA
SCL
I2C
PORT 2
A08
A09
A10
A11
A12
A13
A14
A15
ADDRESS BUS
PORT 4
CMSR0
CMSR1
CMSR2
CMSR3
CMSR4
CMSR5
CMT0
CMT1
TIMER 2 OUTPUTS
LS10940S

C1, C2 = 30 pF ± 10 pF FOR CRYSTALS
= 40 pF ± 10 pF FOR CERAMIC RESONATORS

NOTES:
C1, C2 = 30pfd ± 10pfd. for crystals,
= 40pfd ± 10pfd. for ceramic resonators

I^{2}C LICENSE

Purchase of Signetics or Philips I^{2}C components conveys a license under the Philips I^{2}C patent rights to use these components in an I^{2}C system, provided that the system conforms to the I^{2}C standard specification as defined by Philips.

Signetics

SC80C652/SC83C652
CMOS Single-Chip 8-Bit Microcontroller

Objective Specification

Microprocessor Products

DESCRIPTION

The SC80C652/83C652 is a derivative of the SC8051 family, fabricated with Signetics high-density CMOS technology. Signetics epitaxial substrate minimizes CMOS latch-up sensitivity.

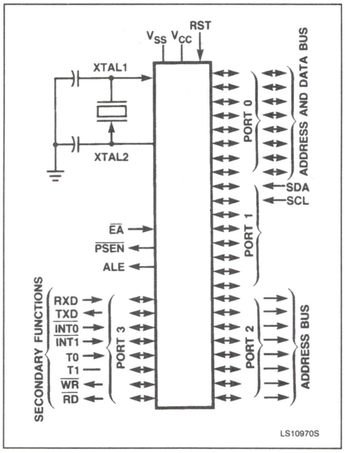
The onboard Inter-Integrated Circuit (I^2C) bus interface allows the SC83C652 to operate as a master or slave device on the I^2C small area network. This capability facilitates I/O and RAM expansion; access to EEPROM; processor-to-processor communication; and efficient interface to a wide variety of dedicated I^2C peripherals.

The SC83C652 contains a 8K × 8 ROM; a 256 × 8 RAM; 32 I/O lines; two 16-bit counter/timers; a six-source, two-priority nested interrupt structure; a byte level bidirectional I^2C serial bus; a full duplex serial UART; on-chip oscillator, and clock circuits.

FEATURES

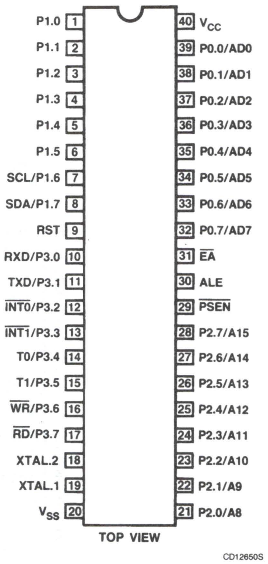
- **Bidirectional I^2C serial bus interface**
- **SC80C51 based architecture**
- **SC80C652 — ROMless version**
- **SC83C652 — Mask programmed 8K ROM version**
- **256 × 8 RAM on-chip**
- **External memory addressing capability:**
  - **64K ROM**
  - **64K RAM**
- **Full duplex serial UART**
- **Two 16-bit counter/timers**
- **Boolean processor**
- **Low power consumption:**
  - **Normal operation: less than 24mA at 5V, 12Mhz**
  - **Idle mode: 3ma typical at 5V, 12Mhz**
  - **Power-down mode: 50μa typical at 2 Volts**
- **CMOS and TTL compatible**
- **Packages:**
  - **40 pin DIP**
  - **44 pin PLCC**

LOGIC SYMBOL

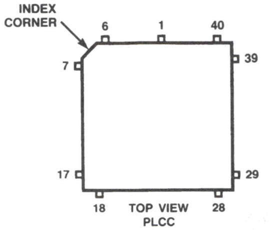


RST
V_SS
V_CC
XTAL1
XTAL2
PORT 0
ADDRESS AND DATA BUS
SDA
SCL
PORT 1
EA
PSEN
ALE
SECONDARY FUNCTIONS
RXD
TXD
INT0
INT1
T0
T1
WR
RD
PORT 3
PORT 2
ADDRESS BUS
LS10970S

PIN CONFIGURATIONS



P1.0 1
P1.1 2
P1.2 3
P1.3 4
P1.4 5
P1.5 6
SCL/P1.6 7
SDA/P1.7 8
RST 9
RXD/P3.0 10
TXD/P3.1 11
INT0/P3.2 12
INT1/P3.3 13
T0/P3.4 14
T1/P3.5 15
WR/P3.6 16
RD/P3.7 17
XTAL.2 18
XTAL.1 19
V_SS 20
40 V_CC
39 P0.0/AD0
38 P0.1/AD1
37 P0.2/AD2
36 P0.3/AD3
35 P0.4/AD4
34 P0.5/AD5
33 P0.6/AD6
32 P0.7/AD7
31 EA
30 ALE
29 PSEN
28 P2.7/A15
27 P2.6/A14
26 P2.5/A13
25 P2.4/A12
24 P2.3/A11
23 P2.2/A10
22 P2.1/A9
21 P2.0/A8
TOP VIEW
CD12650S



INDEX CORNER
6
1
40
7
39
17
29
18
28
TOP VIEW
PLCC
CD12660S

Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	$\overline{\mathrm{PSEN}}$
11	RxD/P3.0	33	ALE
12	NC	34	NC
13	TxD/P3.1	35	$\overline{\mathrm{EA}}$
14	$\overline{\mathrm{INT0}}$/P3.2	36	P0.7/AD7
15	$\overline{\mathrm{INT1}}$P3.3	37	P0.6/AD6
16	T1/P3.4	38	P0.5.AD5
17	T1/P3.5	39	P0.4/AD4
18	$\overline{\mathrm{WR}}$/P3.6	40	P0.3/AD3
19	$\overline{\mathrm{RD}}$/P3.7	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{CC}

CMOS Single-Chip 8-Bit Microcontroller SC80C652/SC83C652

ARCHITECTURE

The SC80C652 is a ROMless version of the SC83C652. This is the only difference between both types; therefore, in the following data wherever the part number SC83C652 is used, the same reference also applies to the part number SC80C652, unless otherwise stated.

This data sheet describes only the added features of the SC83C652, because the SC83C652 is a further development of the SC80C51 microcontroller. It is assumed that the user is familiar with the SC80C51 and its instruction set.

The SC83C652 and SC80C51 have the same instruction set. All new features of the SC83C652 are accessed via additional special function registers.

MEMORY

The SC83C652 contains 8K bytes of program ROM. The SC80C652 is a ROMless version.

The on-chip RAM size for both parts is 256 bytes.

I^2C SERIAL I/O

The I^2C bus consists of two lines; a clock line (SCL), and a data line (SDA). These lines are alternate functions of the port bits P1.6 (SCL) and P1.7 (SDA). These pins are provided with open-drain drivers. They do not have pull-up devices. The input threshold voltage of these two pins is not designed to be compatible to TTL logic levels, but to meet the I^2C signal specification. An input voltage below 1.5V will be recognized as a logic 0, while a voltage above 3V will be a 1.

The I^2C interface stays active in IDLE mode; it can terminate the IDLE mode by generating an interrupt.

Clock generation, address recognition, and arbitration are performed by hardware to reduce software overhead.

The I^2C interface is controlled through registers located in the special function register address range.

I^2C Registers

Mnemonic HEX Function

S1CON	D8	I_2C control[1]
S1STA	D9	I_2C status[2]
S1DAT	DA	I_2C data
S1ADR	DB	I_2C slave addr

NOTES:

1. Write only register
2. Read only register

S1CON Register

MSB 7	6	5	4	3	2	1	LSB 0
X	ENS1	STA	STO	SI	AA	CR1	CR0

DF06660S

The function of the I^2C-SIO section is controlled by the S1CON register. Only two bits are affected by hardware; the serial interrupt bit (SI), and the stop bit (STO).

The SI bit is set to flag an interrupt request.

The STO bit is set by software to generate a stop condition, and is reset when a stop condition is actually detected on the bus.

S1CON.7 — Not used.

S1CON.6 — ENS1, Enable SIO1
When set, the port bits P1.6/P1.7 become SCL/SDA of the I^2C serial bus interface.

S1CON.5 — STA, Start Bit
When this bit is set, the hardware checks the bus and generates a start condition when the bus is free; or a repeated start condition when the bus is not free, and the device is a master/transmitter or a master/receiver.

S1CON.4 — STO, Stop Bit
When set in the master mode, this bit generates a stop condition. When actually detected on the bus, a stop condition clears this bit. When set in the slave mode, no stop condition is generated on the bus, but the device acts as if a stop condition has occurred.

S1CON.3 — SI, SIO1 Interrupt Bit
This bit is set by hardware and an interrupt request is generated, when:

- A start condition is generated in MST mode.
- The microcontroller's own slave address has been received, when AA = 1.
- The general call address has been received, when S1ADR.0 = 1 and AA (S1CON.2) = 1.
- A data byte has been received or transmitted in MST mode, even if arbitration is lost.
- A data byte has been received or transmitted as a selected slave.
- A stop condition is received as a selected slave receiver.

S1CON.2 — AA, Assert Acknowledge
When this bit is set, an acknowledge is returned when either:

- The microcontroller's own slave address is received.
- The general call address is received when S1ADR.0 = 1.
- A data byte is received, when the device is programmed to be a master receiver (MST/REC).
- A data byte is received, while the device is the selected slave receiver (SLV/REC).

When this bit is reset, no acknowledge is returned. No interrupt is requested when the microcontroller's own slave address or the general call address is received.

S1CON.1 — CR1, Clock Rate 1

S1CON.0 — CR0, Clock Rate 0 —
These bits determine the SIO1 clock frequency (fosc/n) in MST mode:

CR1	CR0	n	SCL frequency (fosc = 12MHz
0	0	960	12.5kHz
0	1	120	100kHz
1	0	60	200kHz
1	1	SI01 clock is T1 overflow /8	

S1STA Register

MSB 7	6	5	4	3	2	1	LSB 0
SC4	SC3	SC2	SC1	SC0	0	0	0

DF06670S

The register S1STA is a status register that reflects the state of SIO1 and the I^2C bus. The status is coded in the upper 5 bits of S1STA, while the remaining bits of S1STA are kept low. The contents of this register can be used as a vector into user subroutines, which minimizes the response time of the software and optimizes the utilization of the I^2C bus. S1STAT is a read-only register.

S1DAT Register
S1DAT is the data shift register. It is loaded by software with the data to be transmitted, or holds the data received. Data is shifted in or out MSB first.

S1ADR Register
The upper seven bits of this register are the slave address of the microcontroller when the microcontroller is in the slave receive mode. The device only reacts to the general call address when the LSB of S1ADR is a 1.

INTERRUPTS

The SC80C51 interrupt logic allows a maximum of five interrupts. In the SC83C652 a second layer of interrupt logic is implemented to provide 6 interrupts.

Every interrupt can be individually enabled or disabled by setting a corresponding bit in the IE register. Furthermore, each interrupt can be set to a low or high priority level by setting, or resetting, a bit in the IP register.

IE: Interrupt Enable Register

MSB 7	6	5	4	3	2	1	LSB 0
EA	—	ES1	ES0	ET1	EX1	ET0	EX0

DF06680S

The IE register controls which interrupts are active.

IE.7 EA — Enable All
When this bit is cleared to 0 all interrupts are disabled. When set, each interrupt is enabled individually.

IE.6 — Not used.

IE.5 ES1 — Enable SIO 1 (I^2C) Interrupt

IE.4 ES0 — Enable SIO 0 (UART) Interrupt

IE.3 ET1 — Enable Timer 1 Interrupt

IE.2 EX1 — Enable External 1 Interrupt

IE.1 ET0 — Enable Timer 0 Interrupt

IE.0 EX0 — Enable External 0 Interrupt

Interrupt Priority
Each interrupt source can be provided with one of two priority levels by software. If a level 1 interrupt and a level 0 interrupt are requested at the same time, the processor branches to the level 1 vector.

If there are interrupt requests of sources of the same priority level at the same time, interrupts will be serviced in the following order:

Service Order For Coinciding Interrupts

— — → Service Order — — →
EXTO S1 TO EXT1 t1 S0

IP: Interrupt Priority Register

MSB 7	6	5	4	3	2	1	LSB 0
—	—	PS1	PS0	PT1	PX1	PT0	PX0

DF06690S

The IP register controls the interrupt priority level of the interrupts.

IP.7 — Not used.

IP.6 — Not used.

IP.5 PS1 — SIO 1 Interrupt Priority Level

IP.4 PS0 — SIO 0 Interrupt Priority Level

IP.3 PT1 — Timer 1 Interrupt Priority Level

IP.2 PX1 — External Interrupt 1 Priority Level

IP.1 PT0 — Timer 0 Interrupt Priority Level

IP.0 PX0 — External Interrupt 0 Priority Level

Interrupt Vector Addresses.

SOURCE	VECTOR
External 0	0003h
Timer 0 overflow	000BH
External 1	0013A
Timer 1 overflow	001BH
Serial I/O (UART)	0023H
Serial I/O (I_2C)	002BH

The interrupt vector addresses used in the SC83C652 are identical to those used in the SC80C51 except for the addition of the I^2C interrupt vector address.

I^2C LICENSE

Purchase of Signetics or Philips I^2C components conveys a license under the Philips I^2C patent rights to use these components in an I^2C system, provided that the system conforms to the I^2C standard specification as defined by Philips.

Signetics

SC96AH Series
Single-Chip 16-Bit Microcontrollers

Preliminary Specification

Microprocessor Products

DESCRIPTION

The SC96AH Series microcontrollers consist of a powerful 16-bit CPU tightly coupled with 8K bytes of program memory, 232 bytes of data memory and I/O ports all integrated onto a single piece of silicon. This combination of resources on a single chip represents the highest level of integration in the world of microcontrollers, making possible new applications which required multiple chip solutions in the past.

The CPU features a register-to-register architecture and a dual-operand ALU which supports bit, byte and word operations, as well as 32-bit double words for a subset of the instructions. With a 12MHz clock frequency, the 8096AH (the generic part number used for SC96AH Series microcontrollers) can perform a 16-bit addition in 1μs and a 16×16 multiply or 32/16 divide in 6.25μs. Instruction execution times average 1 to 2μs in typical applications.

Four high-speed trigger inputs can be used to record the times at which external events occur with a resolution of 2μs. Up to six high-speed pulse generator outputs are provided to trigger external events at preset times. These high-speed inputs and outputs supplement five multi-function 8-bit I/O ports. The high-speed output unit can simultaneously perform software timer functions. Up to four such 16-bit timers can be in operation at once in addition to the two 16-bit hardware timers which are on the chip.

Also provided on-chip are a pulse-width modulated output signal, a watchdog timer and a serial port with a dedicated bit-rate generator. Several family members contain an analog-to-digital converter, which converts up to eight analog inputs into 10-bit digital values. Each conversion take 42μs with a 12MHz clock.

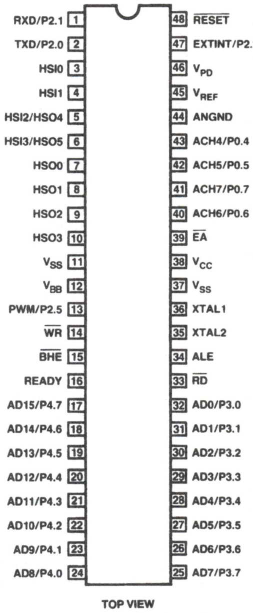
Although 8096AH is the generic part number used throughout this publication, the product family consists of eight part numbers including the 8096AH. This wide variety of products is offered to meet requirements of various users in terms of I/Os and package size. The options include on-board mask programmed memory, 10-bit A/D converter and 48- or 68-pin packages, as shown in the SC96AH family table.

The 8096AH can address up to 64K bytes of memory, so that external memory can be used either as the sole program memory (809X versions) or to expand the 8K bytes of internal program ROM (839X versions). External memory can also be used for data storage.

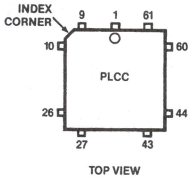
FEATURES

- **Single-chip 16-bit microcontroller**
- **Register to register architecture**
- **8K bytes internal ROM (839XAH versions)**
- **64K byte addressing range**
- **232 byte register file**
  - **16 bytes power-down RAM available**
- **Versatile 2-operand ALU**
  - **Bit, byte, word and double word operands**
  - **Hardware multiply and divide**
  - **1μs basic instruction cycle**
- **10-bit A/D converter**
  - **42μs conversion time**
  - **Up to 8 analog inputs**
- **Pulse width modulated D/A output**
- **Up to five 8-bit I/O ports**
- **High-speed I/O system**
  - **Four inputs and six outputs**
- **Full duplex serial port**
  - **Dedicated bit rate generator**
- **20 interrupt sources**
- **Two 16-bit counter/timers**
- **16-bit watchdog timer**
- **Four 16-bit software timers**
- **48- and 68-pin packages**

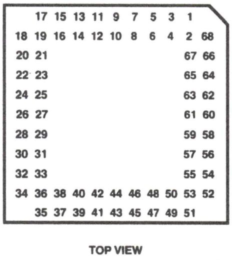
PIN CONFIGURATIONS


RXD/P2.1 1
TXD/P2.0 2
HSI0 3
HSI1 4
HSI2/HSO4 5
HSI3/HSO5 6
HSO0 7
HSO1 8
HSO2 9
HSO3 10
VSS 11
VBB 12
PWM/P2.5 13
WR 14
BHE 15
READY 16
AD15/P4.7 17
AD14/P4.6 18
AD13/P4.5 19
AD12/P4.4 20
AD11/P4.3 21
AD10/P4.2 22
AD9/P4.1 23
AD8/P4.0 24
48 RESET
47 EXTINT/P2.2
46 VPD
45 VREF
44 ANGND
43 ACH4/P0.4
42 ACH5/P0.5
41 ACH7/P0.7
40 ACH6/P0.6
39 EA
38 VCC
37 VSS
36 XTAL1
35 XTAL2
34 ALE
33 RD
32 AD0/P3.0
31 AD1/P3.1
30 AD2/P3.2
29 AD3/P3.3
28 AD4/P3.4
27 AD5/P3.5
26 AD6/P3.6
25 AD7/P3.7
TOP VIEW

CD10712S


INDEX CORNER
9
1
61
10
60
PLCC
26
44
27
43
TOP VIEW

CD09520S


17 15 13 11 9 7 5 3 1
18 19 16 14 12 10 8 6 4 2 68
20 21 67 66
22 23 65 64
24 25 63 62
26 27 61 60
28 29 59 58
30 31 57 56
32 33 55 54
34 36 38 40 42 44 46 48 50 53 52
35 37 39 41 43 45 47 49 51
TOP VIEW

CD10732S

PGA/PLCC PINOUT

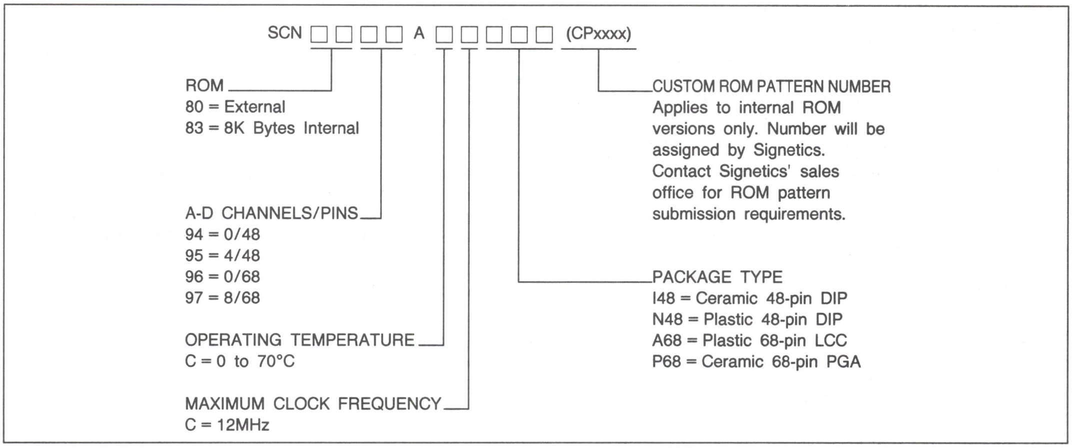
PGA	PLCC	FUNCTION	PGA	PLCC	FUNCTION
1	9	ACH7/P0.7	35	43	READY
2	8	ACH6/P0.6	36	42	T2RST/P2.4
3	7	ACH2/P0.2	37	41	$\overline{BHE}$
4	6	ACH0/P0.0	38	40	$\overline{WR}$
5	5	ACH1/P0.1	39	39	PWM/P2.5
6	4	ACH3/P0.3	40	38	P2.7
7	3	NMI	41	37	V_{BB}
8	2	$\overline{EA}$	42	36	V_{SS}
9	1	V_{CC}	43	35	HSO.3
10	68	V_{SS}	44	34	HSO.2
11	67	XTAL1	45	33	P2.6
12	66	XTAL2	46	32	P1.7
13	65	CLKOUT	47	31	P1.6
14	64	$\overline{TEST}$	48	30	P1.5
15	63	INST	49	29	HSO.1
16	62	ALE	50	28	HSO.0
17	61	$\overline{RD}$	51	27	HSO.5/HSI.3
18	60	AD0/P3.0	52	26	HSO.4/HSI.2
19	59	AD1/P3.1	53	25	HSI.1
20	58	AD2/P3.2	54	24	HSI.0
21	57	AD3/P3.3	55	23	P1.4
22	56	AD4/P3.4	56	22	P1.3
23	55	AD5/P3.5	57	21	P1.2
24	54	AD6/P3.6	58	20	P1.1
25	53	AD7/P3.7	59	19	P1.0
26	52	AD8/P4.0	60	18	TXD/P2.0
27	51	AD9/P4.1	61	17	RXD/P2.1
28	50	AD10/P4.2	62	16	$\overline{RESET}$
29	49	AD11/P4.3	63	15	EXTINT/P2.2
30	48	AD12/P4.4	64	14	V_{PD}
31	47	AD13/P4.5	65	13	V_{REF}
32	46	AD14/P4.6	66	12	ANGND
33	45	AD15/P4.7	67	11	ACH4/P0.4
34	44	T2CLK/P2.3	68	10	ACH5/P0.5

SC96AH FAMILY

FEATURES	PACKAGES	ROMLESS	WITH ROM
Digital I/O only	48-pin	8094AH	8394AH
Digital I/O and 4-channel A/D	48-pin	8095AH	8395AH
Digital I/O only	68-pin	8096AH	8396AH
Digital I/O and 8-channel A/D	68-pin	8097AH	8397AH

Single-Chip 16-Bit Microcontrollers SC96AH Series

ORDERING INFORMATION



SCN ☐☐☐☐ A ☐☐☐☐☐ (CPxxxx)

ROM
80 = External
83 = 8K Bytes Internal

A-D CHANNELS/PINS
94 = 0/48
95 = 4/48
96 = 0/68
97 = 8/68

OPERATING TEMPERATURE
C = 0 to 70°C

MAXIMUM CLOCK FREQUENCY
C = 12MHz

CUSTOM ROM PATTERN NUMBER
Applies to internal ROM versions only. Number will be assigned by Signetics. Contact Signetics' sales office for ROM pattern submission requirements.

PACKAGE TYPE
I48 = Ceramic 48-pin DIP
N48 = Plastic 48-pin DIP
A68 = Plastic 68-pin LCC
P68 = Ceramic 68-pin PGA

PIN DESCRIPTION

SYMBOL	NAME AND FUNCTION
V_{CC}	Main supply voltage (5V).
V_{SS}	Digital circuit ground (0V).
V_{PD}	RAM standby supply voltage (5V). This voltage must be present during normal operation. In a power down condition (i.e. V_{CC} drops to zero), if $\overline{RESET}$ is activated before V_{CC} drops below specification and V_{PD} continues to be held within specification the top 16 bytes in the register file will retain their contents. $\overline{RESET}$ must be held low during the power down and should not be brought high until V_{CC} is within specification and the oscillator has stabilized.
V_{REF}	Reference voltage for the A/D converter (5V). V_{REF} is also the supply voltage to the analog portion of the A/D converter and the logic used to read port 0.
ANGND	Reference ground for the A/D converter. Should be held at nominally the same potential as V_{SS}.
V_{BB}	Substrate voltage from the on-chip back-bias generator. This pin should be connected to ANGND through a 0.01μf capacitor (and not connected to anything else).
XTAL1	Input of the oscillator inverter and of the internal clock generator.
XTAL2	Output of the oscillator inverter.
CLKOUT	Output of the internal clock generator. The frequency of CLKOUT is one-third the oscillator frequency. It has a 33% duty cycle.
$\overline{RESET}$	Reset input to the chip. Input low for at least 2 state times to reset the chip. The subsequent LOW-to-HIGH transition re-synchronizes CLKOUT and commences a 10-state-time sequence in which the PSW is cleared, and a jump to location 2080H is executed. Input high for normal operation. $\overline{RESET}$ has an internal pullup.
$\overline{TEST}$	Input low enables a factory test mode. This pin should be connnected to V_{CC} for normal operation.
NMI	A positive transition clears the watchdog timer count, and causes a vector to external memory location 0000H. External memory from 00H through 0FFH is reserved for development systems.
INST	Output high during an external memory read indicates the read is an instruction fetch. INST is valid throughout the bus cycle.
$\overline{EA}$	Input for memory select (External Access). $\overline{EA}$ equal to a TTL-high causes memory accesses to locations 2000H through 3FFFH to be directed to on-chip ROM. $\overline{EA}$ equal to a TTL-low causes accesses to these locations to be directed to off-chip memory. $\overline{EA}$ has an internal pulldown, so it goes to 0 unless driven otherwise.
ALE	Address Latch Enable output, to demultiplex the address from the address/data bus. ALE is activated only during external memory accesses.
$\overline{RD}$	Read signal output to external memory. $\overline{RD}$ is activated only during external memory reads.
$\overline{WR}$	Write output to external memory. $\overline{WR}$ is activated only during external memory writes.

PIN DESCRIPTION (Continued)

SYMBOL	NAME AND FUNCTION
$\overline{BHE}$	Bus High Enable output to external memory. $\overline{BHE} = 0$ selects the bank of memory that is connected to the high byte of the data bus. A0 = 0 selects the bank of memory that is connected to the low byte of the data bus. Thus, accesses to a 16-bit wide memory can be to the low byte only (A0 = 0, $\overline{BHE} = 1$), to the high byte only (A0 = 1, $\overline{BHE} = 0$), or both bytes (A0 = 0, $\overline{BHE} = 0$). $\overline{BHE}$ is activated only when required during accesses to external memory. $\overline{BHE}$ can be ignored during read operations.
READY	Ready input to lengthen external memory cycles, for interfacing to slow or dynamic memory, or for bus sharing. If the pin is high, CPU operation continues in a normal manner. If the pin is low prior to the falling edge of CLKOUT, the memory controller goes into a wait mode until the next positive transition in CLKOUT occurs with READY high. The bus cycle can be lengthened by up to one μs. When the external memory is not being used, READY has no effect. READY has a weak internal pullup, so it goes to 1 unless externally pulled low.
HSI.0-HSI.3	Inputs to high speed input unit. Four HSI pins are available: HSI.0, HSI.1, HSI.2, and HSI.3. Two of them (HSI.2 and HSI.3) are shared with the HSO unit.
HSO.0-HSO.5	Outputs from high speed output unit. Six HSO pins are available: HSO.0, HSO.1, HSO.2, HSO.3, HSO.4, and HSO.5. Two of them (HSO.4 and HSO.5) are shared with the HSI unit.
ACH0/P0.0- ACH7/P0.7	Port 0. 8-bit high impedance input-only port. These pins can be used as digital inputs and/or as analog inputs to the on-chip A/D converter.
P1.0-P1.7	Port 1. 8-bit quasi-bidirectional I/O port.
P2.0-P2.7	Port 2. 8-bit multi-functional port. Six of its pins (P2.0-P2.5) are shared with other functions, the remaining two are quasi-bidirectional.
AD0/P3.0- AD7/P3.7 and AD8/P4.0- AD15/P4.7	Ports 3 and 4. 8-bit bi-directional I/O ports with open drain outputs. These pins are shared with the multiplexed address/data bus which has strong internal pullups.

NOTE:
The following signals are not provided on the 48-pin components:
ACH0/P0.0 – ACH3/P0.3
P1.0 – P1.7
P2.3, P2.4, P2.6 and P2.7
CLKOUT, INST, NMI and $\overline{TEST}$
The alternate functions on P2.3 (T2CLK) and P2.4 (T2RST) can be programmed to be available on HSI.1 and HSI.0 respectively.

Single-Chip 16-Bit Microcontrollers SC96AH Series

FUNCTIONAL DESCRIPTION

The 8096AH can be separated into several sections for the purpose of describing its operation. Featured is a CPU, a programmable high speed I/O unit, an analog to digital converter, a serial port, and a pulse width modulated (PWM) output for digital to analog conversion. In addition to these functional units, there are some sections which support overall operation of the chip. A block diagram of the 8096AH is illustrated in Figure 1.

CPU Architecture

The major components of the CPU are the register file and the register/arithmetic logic unit (RALU).

The 8096AH uses the same address space for both program and data memory, except in the address range from 00H through 0FFH. Data fetches in this range are always to the register file, while instruction fetches from these locations are directed to external memory. (Locations 00H through 0FFH in external memory are reserved for development systems).

Within the register file, locations 00H through 17H are register mapped I/O control registers, also referred to as special function registers (SFRs). The rest of the register file (018H through 0FFH) contains 232 bytes of RAM, which can be referenced as bytes, words, or double-words. This register space allows the user to keep the most frequently-used variables in on-chip RAM, which can be accessed faster than external memory. Locations 0F0H through 0FFH can be preserved during power down via a separate power down pin (V_{PD}).

Locations 18H and 19H contain the stack pointer. These are not SFRs, and may be used as standard RAM if stack operations are not being performed. The stack pointer must be initialized by the user program and can point anywhere in the 64K memory space. The stack, however, builds down. There are no restrictions on the use of the remaining 230 locations except that code cannot be executed from them.

Outside of the register file, program memory, data memory, and peripherals can be intermixed. The addresses with special significance are:

0000H-	0017H	Register Mapped I/O (SFRs)
0018H-	0019H	Stack Pointer
1FFEH-	1FFFH	Ports 3 and 4
2000H-	2011H	Interrupt Vectors
2012H-	207FH	Factory Test Code
2080H		Reset Location

The 839XAH carries 8K bytes of ROM for program memory which occupies addresses 2000H through 3FFFH. Instruction or data fetches from these addresses access the on-chip memory if the $\overline{EA}$ pin is externally held at a logical 1. If the $\overline{EA}$ pin is at a logical 0, these addresses access off-chip memory.

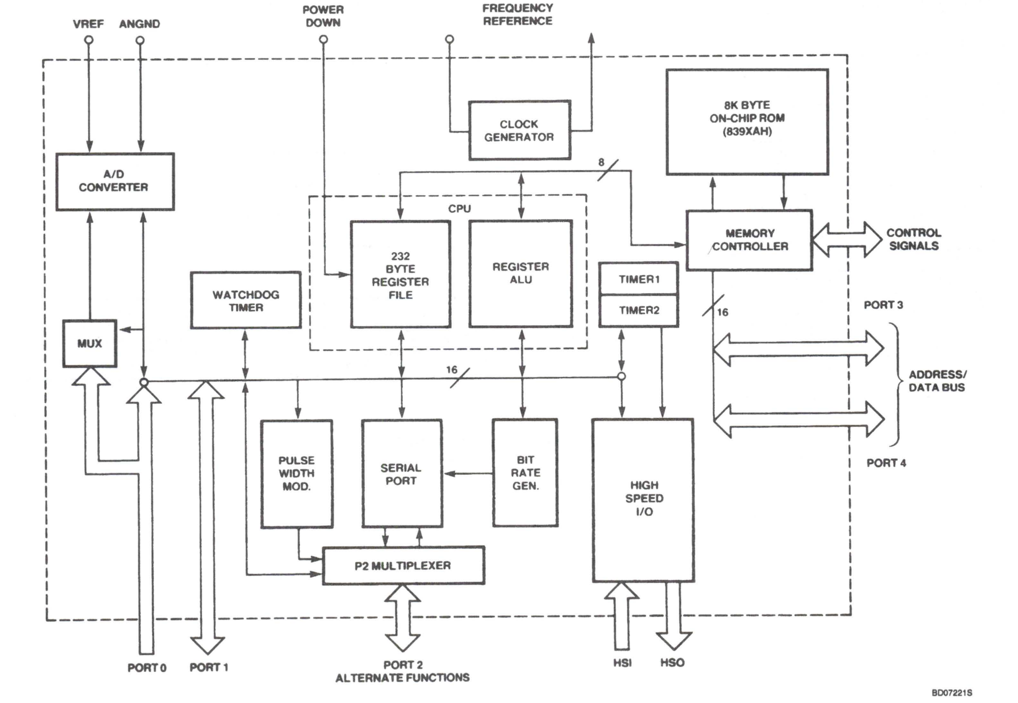

VREF
ANGND
POWER DOWN
FREQUENCY REFERENCE
CLOCK GENERATOR
8K BYTE ON-CHIP ROM (839XAH)
A/D CONVERTER
8
CPU
232 BYTE REGISTER FILE
REGISTER ALU
MEMORY CONTROLLER
CONTROL SIGNALS
WATCHDOG TIMER
TIMER1
TIMER2
16
PORT 3
MUX
16
ADDRESS/ DATA BUS
PORT 4
PULSE WIDTH MOD.
SERIAL PORT
BIT RATE GEN.
HIGH SPEED I/O
P2 MULTIPLEXER
PORT 0
PORT 1
PORT 2 ALTERNATE FUNCTIONS
HSI
HSO
BD07221S

Figure 1. SC96AH Series Block Diagram

A memory map for the SC96AH product family is shown in Figure 2. Table 1 summarizes the function of each SFR. Note that these registers can be accessed only as bytes unless noted otherwise.

The RALU section consists of a 17-bit ALU, the program status word, the program counter, and several temporary registers. A key feature of the 8096AH is that it does not use an accumulator. Rather, it operates directly on any register in the register file. Being able to operate directly on data in the register file without having to move it into and out of an accumulator results in a significant improvement in execution speed. A control unit and two buses connect the register file and RALU. The two buses are the "A-Bus" which is 8-bits wide, and the "D-Bus" which is 16-bits wide. The D-Bus transfers data only between the RALU and the register file or SFRs. The A-Bus is used as the address bus for the above transfers or as a multiplexed address/data bus connecting to the memory controller. Any accesses of either the internal ROM or external memory are done through the memory controller.

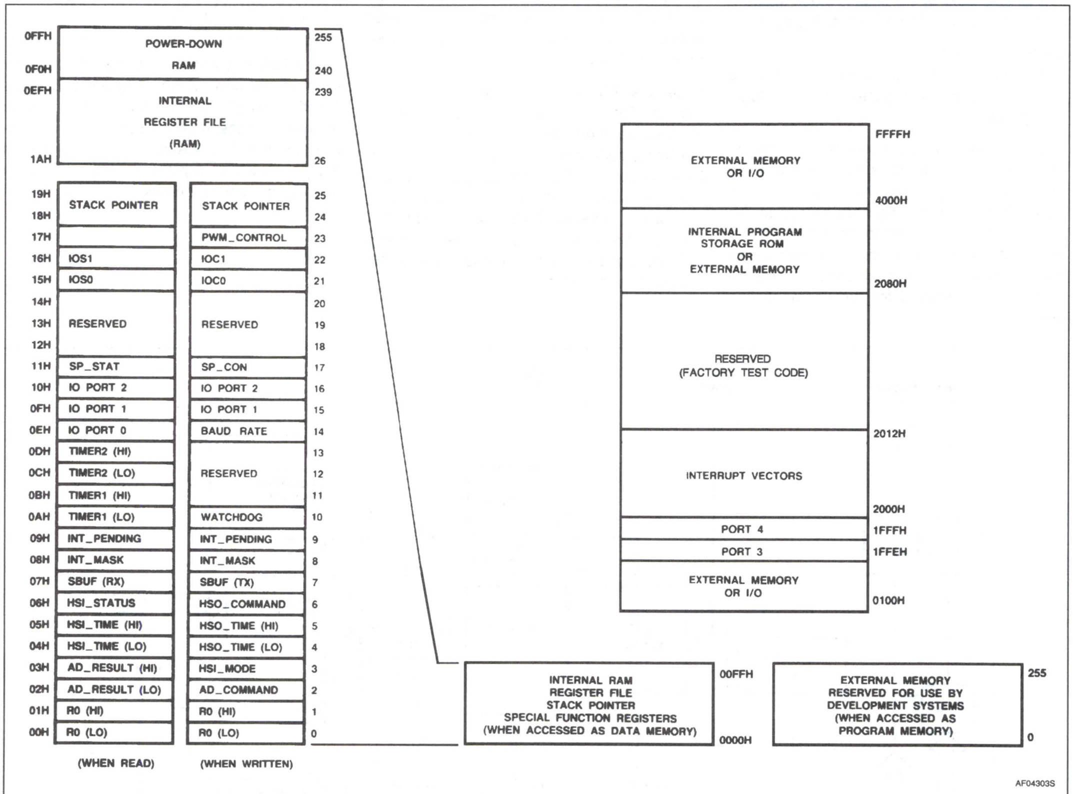

0FFH
POWER-DOWN
RAM
255
0F0H
240
0EFH
239
INTERNAL
REGISTER FILE
(RAM)
1AH
26
19H
18H
STACK POINTER
STACK POINTER
25
24
17H
PWM_CONTROL
23
16H
IOS1
IOC1
22
15H
IOS0
IOC0
21
14H
20
13H
RESERVED
RESERVED
19
12H
18
11H
SP_STAT
SP_CON
17
10H
IO PORT 2
IO PORT 2
16
0FH
IO PORT 1
IO PORT 1
15
0EH
IO PORT 0
BAUD RATE
14
0DH
TIMER2 (HI)
13
0CH
TIMER2 (LO)
RESERVED
12
0BH
TIMER1 (HI)
11
0AH
TIMER1 (LO)
WATCHDOG
10
09H
INT_PENDING
INT_PENDING
9
08H
INT_MASK
INT_MASK
8
07H
SBUF (RX)
SBUF (TX)
7
06H
HSI_STATUS
HSO_COMMAND
6
05H
HSI_TIME (HI)
HSO_TIME (HI)
5
04H
HSI_TIME (LO)
HSO_TIME (LO)
4
03H
AD_RESULT (HI)
HSI_MODE
3
02H
AD_RESULT (LO)
AD_COMMAND
2
01H
R0 (HI)
R0 (HI)
1
00H
R0 (LO)
R0 (LO)
0
(WHEN READ)
(WHEN WRITTEN)
EXTERNAL MEMORY
OR I/O
FFFFH
4000H
INTERNAL PROGRAM
STORAGE ROM
OR
EXTERNAL MEMORY
2080H
RESERVED
(FACTORY TEST CODE)
2012H
INTERRUPT VECTORS
2000H
PORT 4
1FFFH
PORT 3
1FFEH
EXTERNAL MEMORY
OR I/O
0100H
INTERNAL RAM
REGISTER FILE
STACK POINTER
SPECIAL FUNCTION REGISTERS
(WHEN ACCESSED AS DATA MEMORY)
00FFH
0000H
EXTERNAL MEMORY
RESERVED FOR USE BY
DEVELOPMENT SYSTEMS
(WHEN ACCESSED AS
PROGRAM MEMORY)
255
0
AF04303S

Figure 2. Memory Map

Table 1. Special Function Registers

REGISTER	DESCRIPTION
R0	Zero Register — Always reads as a zero, useful for a base when indexing and as a constant for calculations and compares.
AD_RESULT	A/D Result Hi/Low — Low and high order results of the A/D converter (byte read only).
AD_COMMAND	A/D Command Register — Controls the A/D.
HSI_MODE	HSI Mode Register — Sets the mode of the high speed input unit.
HSI_TIME	HSI Time Hi/Lo — Contains the time at which the high-speed input unit was triggered (word read only).
HSO_TIME	HSO Time Hi/Lo — Sets the time for the high speed output to execute the command in the command register (word write only).
HSO_COMMAND	HSO Command Register — Determines what will happen at the time loaded into the HSO time registers.
HSI_STATUS	HSI Status Registers — Indicates which HSI pins were detected at the time in the HSI time registers.
SBUF (TX)	Transmit buffer for the serial port, holds contents to be outputed.
SBUF (RX)	Receive buffer for the serial port, holds the byte just received by the serial port.
INT_MASK	Interrupt Mask Register — Enables or disables the individual interrupts.
INT_PENDING	Interrupt Pending Register — Indicates when an interrupt signal has occurred on one of the sources.
WATCHDOG	Watchdog Timer Register — Written to periodically to hold off automatic reset every 64K state times.
TIMER1	Timer 1 Hi/Lo — Timer 1 high and low bytes (word read only).
TIMER2	Timer 2 Hi/Lo — Timer 2 high and low bytes (word read only).
IOPORT0	Port 0 Register — Levels on pins of port 0.
BAUD_RATE	Register which contains the baud rate. This register is loaded sequentially.
IOPORT1	Port 1 Register — Used to read or write to port 1.
IOPORT2	Port 2 Register — Used to read or write to port 2.
SP_STAT	Serial Port Status — Indicates the status of the serial port.
SP_CON	Serial port control — Used to set the mode of the serial port.
IOS0	I/O Status Register 0 — Contains information on the HSO status.
IOS1	I/O Status Register 1 — Contains information on the status of the timers and of the HSI.
IOC0	I/O Control Register 0 — Controls alternate functions of HSI pins, timer 2 reset sources and timer 2 clock sources.
IOC1	I/O Control Register 1 — Controls alternate functions of port 2 pins, timer interrupts and HSI interrupts.
PWM_CONTROL	Pulse Width Modulation Control Register — Sets the duration of the PWM pulse.

Timing

The 8096AH requires a clock source to operate. This clock can be provided to the chip through the XTAL1 input or by use of the on-chip oscillator. The frequency of operation is from 6MHz to 12MHz.

The on-chip circuitry for the oscillator is a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator with external connections as shown in Figure 3. In this application, the crystal is being operated in its fundamental response mode as an inductive reactance in parallel resonance with capacitance external to the crystal.

The crystal specifications and capacitance values (C1 and C2 in Figure 3) are not critical. With good quality crystals, 30pF can be used in these positions at any frequency. For 0.5% frequency accuracy, the crystal frequency can be specified at series resonance or for parallel resonance with any load capacitance. (In other words, for that degree of frequency accuracy, the load capacitance simply doesn't matter.) For 0.05% frequency accuracy the crystal frequency should be specified for parallel resonance with 25pF load capacitance, if C1 and C2 are 30pF.

To drive the 8096AH with an external clock source, apply the external clock signal to XTAL1 and let XTAL2 float. An example of this circuit is shown in Figure 4. The required voltage levels on XTAL1 are specified in the specifications section of this data sheet. The signal on XTAL1 must be clean with good solid levels. It is important that the minimum high and low times are met.

There is no specification on rise and fall times, but they should be reasonably fast (on the order of 20ns) to avoid having the XTAL1 pin in the transition range for long periods of time. The longer the signal is in the transition region, the higher the probability that an external noise glitch could be seen by the clock generator circuitry, which may result in unreliable operation.

The crystal or external oscillator frequency is divided by three to generate three internal timing phases which repeat every 3 oscillator periods. This is referred to as one "state time", the basic time measurement for 8096AH operations. Most internal operations are synchronized to either phase A, B or C, each of which have a 33% duty cycle. Phase A is represented externally by CLKOUT, a signal available on the 68-pin parts. With a 12MHz clock, each state time requires 0.25μs.

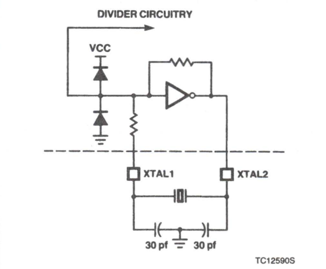


Figure 3. Crystal Oscillator Circuit

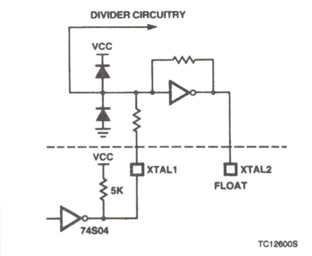


Figure 4. External Clock Drive

Interrupt Structure

Interrupt Sources

Eight interrupt sources are available on the 8096AH. When enabled, an interrupt occurring on any of these sources will force a call to the location stored in the vector location for that source. The interrupt sources and their respective vector locations and priorities are listed in Table 2. In addition to the 8 standard interrupts, there is a TRAP instruction which acts as a software generated interrupt. This instruction is reserved for use by development systems. Many of the interrupt sources can be activated by several methods. Figure 5 shows all of the possible sources for interrupts.

Interrupt Control

A block diagram of the interrupt system is shown in Figure 6. Each of the interrupt sources is tested for a 0 to 1 transition. If this transition occurs, the corresponding bit in the interrupt pending register, located at 0009H, is set. The bit is cleared when the vector is taken to the interrupt routine. Since this register can be written to, it is possible to generate software interrupts by setting bits within the register, or remove pending interrupts by clearing the bits in this register. The pending register can be set even if the interrupt is disabled.

Caution must be used when writing to the pending register to clear interrupts. If the interrupt has already been acknowledged when the bit is cleared, a 4 state-time "partial" interrupt cycle will occur. This is because the 8096AH will have to fetch the next instruction of the normal instruction flow, instead of proceeding with the interrupt processing as it was going to. The effect on the program will be essentially that of an extra NOP. This can be prevented by clearing the bits using a 2-operand immediate logical, as the 8096AH holds off acknowledging interrupts during these "read/modify/write" instructions.

Table 2. Interrupt Vectors and Priorities

SOURCE	VECTOR LOCATION		PRIORITY
	(High Byte)	(Low Byte)	
Software	2011H	2010H	Not Applicable
Extint	200FH	200EH	7 (Highest)
Serial Port	200DH	200CH	6
Software Timers	200BH	200AH	5
HSI.0	2009H	2008H	4
High Speed Outputs	2007H	2006H	3
HSI Data Available	2005H	2004H	2
A/D Conversion Complete	2003H	2002H	1
Timer Overflow	2001H	2000H	0 (Lowest)

Enabling and disabling of individual interrupts is done through the interrupt mask register, located at 0008H. If the bit in the mask register is a 1 then the interrupt is enabled, otherwise it is disabled. Even if an interrupt is masked it may still become pending. It may, therefore, be desirable to clear the pending bit before unmasking an interrupt.

The interrupt mask register is also the low byte of the PSW. All of the interrupts may be enabled and disabled simultaneously by using the "EI" (Enable Interrupt) and "DI" (Disable Interrupt) instructions. EI and DI set and clear PSW.9, the interrupt enable bit, but do not affect the contents of the mask register.

Interrupt Priority Programming

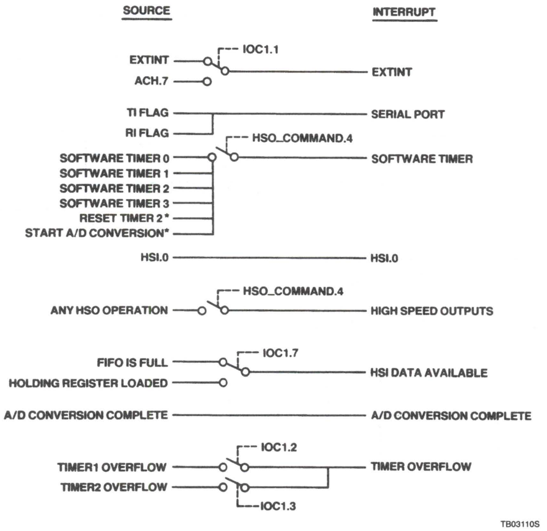
The priority encoder looks at all of the interrupts which are *both pending and enabled*, and selects the one with the highest priority. The interrupt generator then forces a call to the location in the indicated vector location. This location would be the starting location of the interrupt service routine (ISR).

At least one instruction in the ISR (usually "DI" or "PUSHF" [Push Flags]) will always be executed before another interrupt can be acknowledged. The PUSHF instruction pushes the PSW onto the stack and then clears it.

Clearing the PSW disables all interrupts in two ways: first, by clearing PSW.9 (the interrupt enable bit) and second, because the interrupt mask register is located in bits 0 through 7 of the PSW. The interrupts which should be permitted to interrupt this ISR can then be set in the mask register and an "EI" instruction executed.

By selectively determining which interrupts are enabled or disabled within which interrupt service routines, it is possible to configure the interrupt system in any way one would desire.

The last two instructions in an ISR are normally a "POPF" (pop flags), which restores the PSW (and, therefore, the interrupt mask register) followed by a 'RET', which restores the program counter. Execution will then continue from the point at which the call was forced.


SOURCE
INTERRUPT
IOC1.1
EXTINT
ACH.7
EXTINT
TI FLAG
RI FLAG
SERIAL PORT
HSO_COMMAND.4
SOFTWARE TIMER 0
SOFTWARE TIMER 1
SOFTWARE TIMER 2
SOFTWARE TIMER 3
RESET TIMER 2*
START A/D CONVERSION*
SOFTWARE TIMER
HSI.0
HSI.0
HSO_COMMAND.4
ANY HSO OPERATION
HIGH SPEED OUTPUTS
IOC1.7
FIFO IS FULL
HOLDING REGISTER LOADED
HSI DATA AVAILABLE
A/D CONVERSION COMPLETE
A/D CONVERSION COMPLETE
IOC1.2
TIMER1 OVERFLOW
TIMER2 OVERFLOW
TIMER OVERFLOW
IOC1.3
TB03110S

NOTE:
*Only when initiated by the HSO unit.

Figure 5. All Possible Interrupt Sources

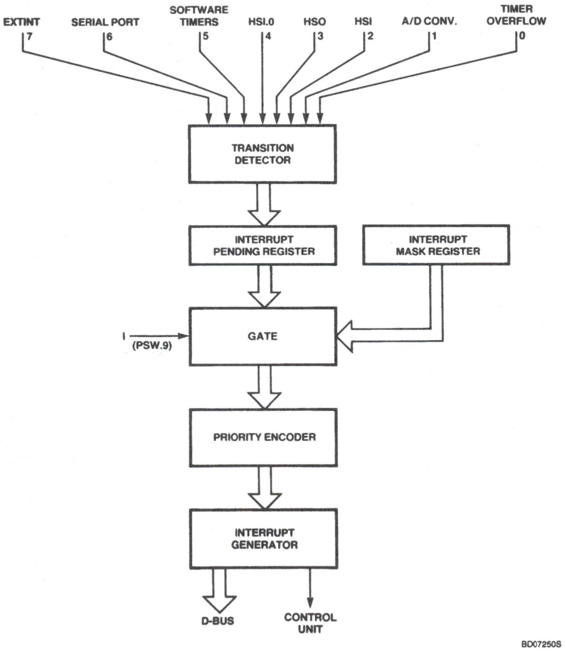

EXTINT 7
SERIAL PORT 6
SOFTWARE TIMERS 5
HSI.0 4
HSO 3
HSI 2
A/D CONV. 1
TIMER OVERFLOW 0
TRANSITION DETECTOR
INTERRUPT PENDING REGISTER
INTERRUPT MASK REGISTER
I (PSW.9)
GATE
PRIORITY ENCODER
INTERRUPT GENERATOR
D-BUS
CONTROL UNIT
BD07250S

Figure 6. Block Diagram of Interrupt System

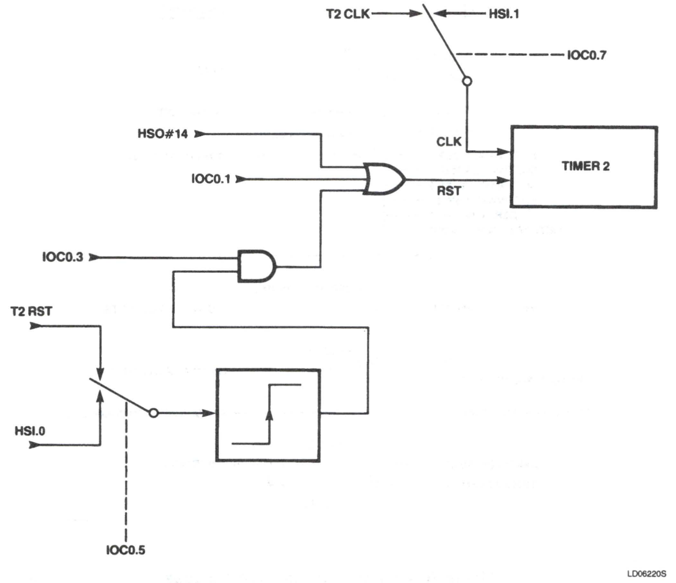


Figure 7. Timer 2 Clock and Reset Options

Timers

Two 16-bit hardware timers are available for use on the 8096AH. Timer 1 is used to synchronize events to real time, while timer 2 can be clocked externally and synchronizes events to external occurrences.

Timer 1 is clocked once every eight state times and can be cleared only by executing a reset. The only other way to change its value is by writing to 000CH but this is a test mode which sets both timers to 0FFFXH and should not be used in programs.

Timer 2 can be incremented by transitions (one count each transition, rising *and* falling) on either T2CLK or HSI.1. The multiple functionality of the timer is determined by the state of I/O control register 0, bit 7 (IOC0.7). To ensure that all CAM entries are checked each count of timer 2, the maximum transition speed is limited to once per eight state times. Timer 2 can be cleared by executing a reset; by setting IOC0.1; by triggering HSO channel 0EH; or by pulling T2RST or HSI.0 high. (The HSO and CAM are described in later sections). IOC0.3 and IOC0.5 control the resetting of timer 2. Figure 7 shows the different ways of manipulating timer 2.

Timer Interrupts

Both timer 1 and timer 2 can be used to trigger a timer overflow interrupt and set a flag in the I/O status register 1 (IOS1). The interrupts are controlled by IOC1.2 and IOC1.3 respectively. The flags are set in IOS1.5 and IOS1.4, respectively.

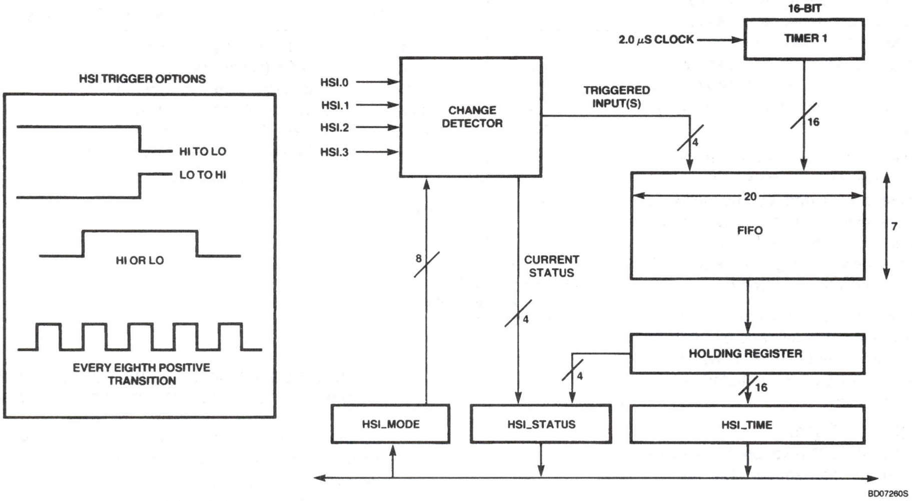
Caution must be used when examining the flags, as any access (including Compare and Jump on Bit) of IOS1 clears the whole byte, including the software timer flags. It is, therefore, recommended to write the byte to a temporary register before testing bits. The general enabling and disabling of the timer interrupts are controlled by the interrupt mask register bit 0. In all cases, setting a bit enables a function, while clearing a bit disables it.

High Speed I/O Unit (HSIO)

The HSIO unit consists of the high speed input unit (HSI) and the high speed output unit (HSO). "High speed" denotes that the units can perform functions related to the timers without CPU intervention. The HSI records the value on timer 1 when transitions occur and the HSO causes transitions to occur based on values of either timer 1 or timer 2.

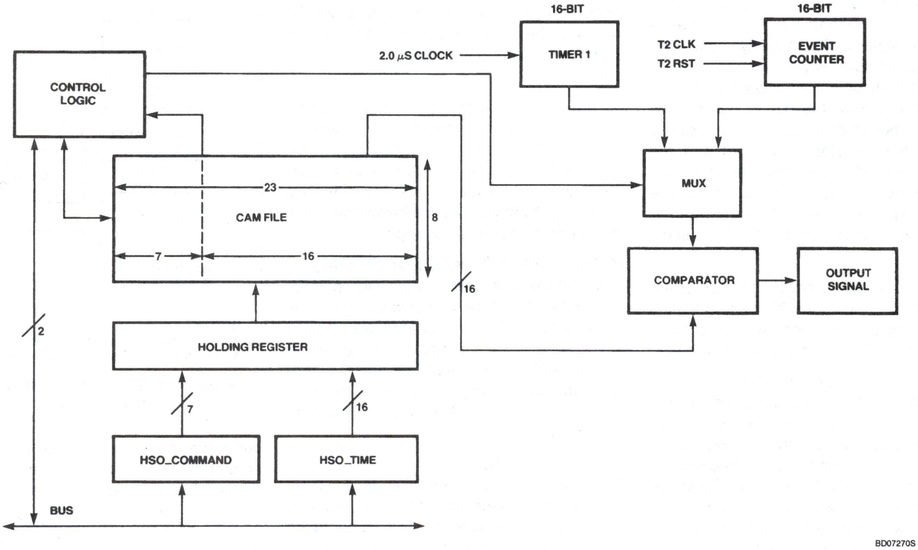
The high speed input unit can detect transitions on any of its four input lines. When one occurs, it records both the time (from timer 1) and which input lines made the transition. This information is recorded with 2μs (12MHz system) resolution and stored in an 8-level FIFO. The unit can be programmed to look for four types of events, as shown in Figure 8. It can activate the HSI data available interrupt either when the holding register is loaded or when the sixth FIFO entry has been made. Each input line can be individually enabled or disabled to the HSI unit by software.

The high speed output unit is shown in Figure 9. It can be programmed to set or clear any of its six output lines, reset timer 2, trigger an A/D conversion or set one of four software timer flags at a programmed time. An interrupt can be enabled for any of these events. Either timer 1 or timer 2 can be referenced for the programmed time value and up to 8 commands for preset actions can be stored in the CAM (content addressable memory) file at any one time. As each action is carried out at its preset time, that command is removed from the CAM making space for another command. HSO.4 and HSO.5 are shared with the HSI unit as HSI.2 and HSI.3, and can be individually enabled or disabled as outputs.


HSI TRIGGER OPTIONS
HI TO LO
LO TO HI
HI OR LO
EVERY EIGHTH POSITIVE TRANSITION
HSI.0
HSI.1
HSI.2
HSI.3
CHANGE DETECTOR
2.0 μS CLOCK
16-BIT
TIMER 1
TRIGGERED INPUT(S)
4
16
20
FIFO
7
8
CURRENT STATUS
4
HOLDING REGISTER
4
16
HSI_MODE
HSI_STATUS
HSI_TIME
BD07260S

NOTES:
- Pulse measurement with 2.0 μsec resolution
- Input transitions trigger the recording of the reference timer (16-bit) and triggered input(s) (4-bit)

Figure 8. High Speed Input Unit


16-BIT
16-BIT
2.0 μS CLOCK
TIMER 1
T2 CLK
T2 RST
EVENT COUNTER
CONTROL LOGIC
MUX
23
CAM FILE
8
7
16
16
COMPARATOR
OUTPUT SIGNAL
2
HOLDING REGISTER
7
16
HSO_COMMAND
HSO_TIME
BUS
BD07270S

NOTES:
High speed output controls
6 pins
4 software timers
2 interrupts
Initiate A/D conversion
Reset timer 2

Figure 9. High Speed Output Unit

Watchdog Timer

The watchdog timer is a 16-bit counter which, once started, is incremented every state time. If not cleared before it overflows, the $\overline{\text{RESET}}$ pin will be pulled down for two state times, causing the system to be reinitialized. In a 12MHz system, the watchdog timer overflows after 16ms.

This feature is provided as a means of graceful recovery from a software upset. The counter must be cleared by the software before it overflows, or else the system assumes an upset has occurred and activates $\overline{\text{RESET}}$. Once the watchdog timer is started it cannot be turned off by software.

To start the watchdog timer (or to clear it), the system user writes 1EH followed by 0E1H to the WDT address (000AH). The Watchdog cannot be stopped once it is started unless the system is reset.

Standard I/O Ports

There are five 8-bit I/O ports on the 8096AH in addition to the high speed I/O lines.

Port 0 is an input-only port which shares its pins with the analog inputs to the A/D converter. The port can be read digitally, and/or by writing to the A/D command register one of the lines can be selected as the input to the A/D converter.

Port 1 is a quasi-bidirectional I/O port. "Quasi-bidirectional" means the port pin has a weak internal pullup that is always active and an internal pulldown which can either be on (to output a 0) or off (to output a 1). This configuration allows the pin to be used as either an input or an output without using a data direction register. In parallel with the weak internal pullup is a much stronger internal pullup that is activated for one state time when the pin is internally driven from 0 to 1. This is done to speed up the 0-to-1 transition time.

Port 2 is a multi-functional port as two of the pins (P2.6, 2.7) are quasi-bidirectional while the remaining six are shared with other functions in the 8096AH, as shown in Table 3.

Ports 3 and 4 are bi-directional I/O ports with open drain outputs. These pins are also used as the multiplexed address/data bus when accessing external memory, in which case they have strong internal pullups. The internal pullups are only used during external memory read or write cycles when the pins are outputting address or data bits. At any other time, the internal pullups are disabled.

Serial Port

The serial port is compatible with the serial port of 8051-type microcontrollers such as the 8031, 8051 and 80C451. It is full duplex, and double-buffered on receive. There are three asynchronous modes and one synchronous mode of operation for the serial port. The asynchronous modes allow for eight or nine bits of data with even parity optionally inserted for one of the data bits. Selective interrupts based on the ninth data bit are available to support interprocessor communication.

Table 3. Port 2 Pin Functions

PORT	FUNCTION	ALTERNATE FUNCTION
P2.0	output	TXD (serial port transmit)
P2.1	input	RXD (serial port receive)
P2.2	input	EXTINT (external interrupt)
P2.3	input	T2CLK (Timer 2 clock)
P2.4	input	T2RST (Timer 2 reset)
P2.5	output	PWM (pulse-width modulation)

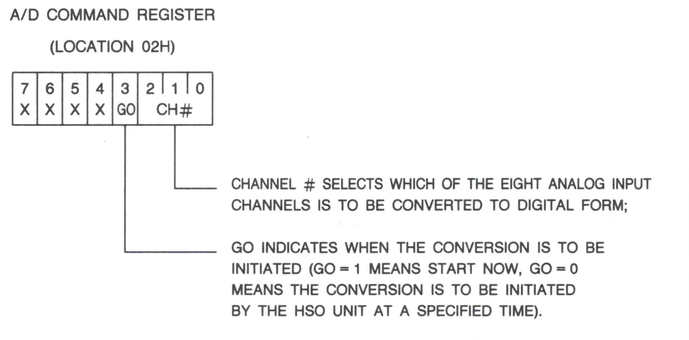

A/D COMMAND REGISTER
(LOCATION 02H)
7 6 5 4 3 2 1 0
X X X X GO CH#
CHANNEL # SELECTS WHICH OF THE EIGHT ANALOG INPUT CHANNELS IS TO BE CONVERTED TO DIGITAL FORM;
GO INDICATES WHEN THE CONVERSION IS TO BE INITIATED (GO = 1 MEANS START NOW, GO = 0 MEANS THE CONVERSION IS TO BE INITIATED BY THE HSO UNIT AT A SPECIFIED TIME).

Figure 11. A/D Command Register

Bit rates in all modes are determined by an independent 16-bit on-chip bit rate generator. Either the XTAL1 pin or the T2CLK pin can be used as the input to the bit rate generator. The maximum bit rate in the asynchronous mode is 187.5Kbps. The maximum rate in the synchronous mode is 1.5Mbps.

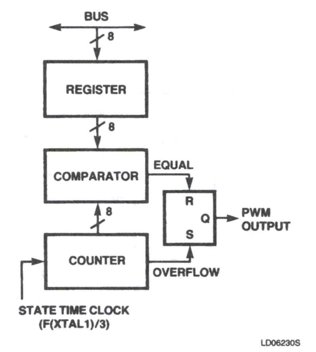
Pulse Width Modulator (PWM)

The PWM output shares a pin with port bit P2.5. When the PWM output is selected, this pin outputs a pulse train having a fixed period of 256 state times, and a programmable width of 0 to 255 state times. The width is programmed by loading the desired value, in state times, to the PWM control register (see Figure 10).

A/D Converter

The analog-to-digital converter is a 10-bit, successive approximation converter. It has a fixed conversion time of 168 state times (42μs with a 12MHz clock). A 2pF capacitance is charged from the input signal during acquisition.

The analog input must be in the range of 0 to V_{REF} (nominally, V_{REF} = 5V). This input can be selected from eight analog input lines, which connect to the same pins as port 0. A conversion can be initiated either by setting a control bit in the A/D command register, Figure 11, or by programming the HSO unit to trigger the conversion at some specified time. The result of the conversion is stored in the A/D result register, Figure 12.



NOTES:
- Period = 64μs, Frequency = 15.625kHz
- Duty factor programmable in 256 steps

Figure 10. Pulse Width Modulated (D/A) Output

RESET

As with all processors, the 8096AH must be reset each time the power is turned on. To complete a reset, the $\overline{\text{RESET}}$ pin must be

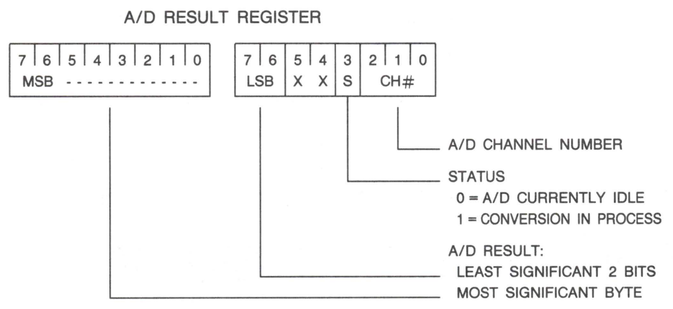


Figure 12. A/D Result Register

held active (low) for at least 2 state times after V_{CC}, the oscillator, and the back bias generator have stabilized (~ 1.0 milliseconds). Then when $\overline{RESET}$ is brought high again, the chip executes a reset sequence that takes 10 state times. This sequence initializes some registers, clears the PSW, and jumps to address 2080H.

The 8096AH can be reset using a capacitor, 1-shot, or any other method capable of providing a pulse of at least two state times longer than required for V_{CC} and the oscillator to stabilize.

For best functionality, it is suggested that the reset pin be pulled low with an open collector device. In this way, several reset sources can be wire-Ored together. Remember that the $\overline{RESET}$ pin itself can be a reset source (see watchdog timer section).

With V_{CC} and the oscillator stabilized, the I/O lines and control lines of the 8096AH will be in their reset state within two state times after reset is low. Prior to that time, the status of the I/O lines is indeterminate. After the 10 state-time reset sequence, the special function registers will be set as follows:

REGISTER	RESET VALUE
Port 1	11111111B
Port 2	110XXXX1B
Port 3	11111111B
Port 4	11111111B
PWM Control	00H
Serial Port (Transmit)	undefined
Serial Port (Receive)	undefined
Baud Rate Register	undefined
Serial Port Control	XXXX0XXXB
Serial Port Status	X00XXXXXB
A/D Command	undefined
A/D Result	undefined
Interrupt Pending	undefined
Interrupt Mask	00000000B
Timer 1	0000H
Timer 2	0000H
Watchdog Timer	0000H
HSI Mode	11111111B
HSI Status	undefined
IOS0	00000000B
IOS1	00000000B
IOC0	X0X0X0X0B
IOC1	X0X0XXX1B
HSI FIFO	empty
HSO CAM	empty
HSO lines	000000B
PSW	0000H
Stack Pointer	undefined
Program Counter	2080H

Other conditions following a reset are:

PIN	RESET VALUE
$\overline{RD}$	high
$\overline{WR}$	high
ALE	low
$\overline{BHE}$	low
INST	high

It is important to note that the stack pointer and interrupt pending register are undefined, and need to be initialized in software. The interrupts are disabled by both the mask register and PSW.9 after a reset.

POWER DOWN

The upper 16 RAM locations (0F0H through 0FFH) receive their power from the V_{PD} pin. If it is desired to keep the memory in these locations alive during a power down situation, one need only keep voltage on the V_{PD} pin. The current required to keep the RAM alive is approximately 1 milliamp (refer to the DC specifications). Both V_{CC} and V_{PD} must have power applied for normal operation.

To place the 8096AH into a power down mode, the $\overline{RESET}$ pin is pulled low. Two state times later the part will be in reset. This is necessary to prevent the part from writing into RAM as the power goes down. The power may now be removed from the V_{CC} pin, the V_{PD} pin must remain within specifications. The 8096AH can remain in this state for any amount of time and the 16 RAM bytes will retain their values.

To bring the 8096AH out of power down, $\overline{RESET}$ is held low while V_{CC} is applied. Two state times after the oscillator and the back bias generator have stabilized (~ 1 millisecond), the $\overline{RESET}$ pin can be pulled high. The 8096AH will begin to execute code at location 02080H 10 state times after $\overline{RESET}$ is pulled high. Figure 13 shows a timing diagram of the power down sequence. To ensure that the 2 state-time minimum reset time (synchronous with CLKOUT) is met, it is recommended that 10 XTAL1 cycles be used.

INSTRUCTION SET

The 8096AH instruction set makes use of six addressing modes as described below:

DIRECT — The operand is specified by an 8-bit address field in the instruction. The operand must be in the register file or SFR space (locations 0000H through 00FFH).

IMMEDIATE — The operand itself follows the opcode in the instruction stream as immediate data. The immediate data can be either 8-bits or 16-bits as required by the opcode.

INDIRECT — An 8-bit address field in the instruction gives the word address of a word register in the register file which contains the 16-bit address of the operand. The operand can be anywhere in memory.

INDIRECT WITH AUTO-INCREMENT — Same as Indirect, except that, after the operand is referenced, the word register that contains the operand's address is incremented by one if the operand is a byte, or by two if the operand is a word.

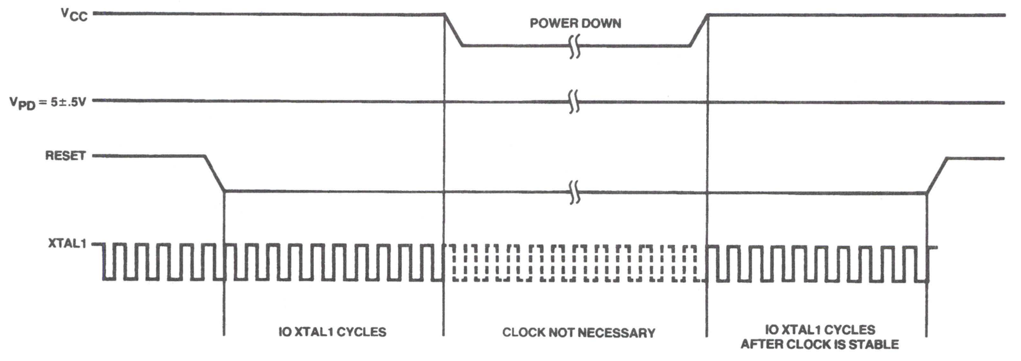


Figure 13. Power Down Timing

INDEXED (LONG AND SHORT) — The instruction contains an 8-bit address field and either an 8-bit or a 16-bit displacement field. The 8-bit address field gives the word address of a word register in the register file which contains a 16-bit base address. The 8- or 16-bit displacement field contains a signed displacement that will be added to the base address to produce the address of the operand. The operand can be anywhere in memory.

The 8096AH contains a Zero Register at word address 0000H (and which contains 0000H). This register is available for performing comparisons and for use as a base register in indexed addressing. This effectively provides direct addressing to all 64K of memory.

In the 8096AH, the stack pointer is at word address 0018H in the register file. If the 8-bit address field contains 18H, the stack pointer becomes the base register. This allows direct accessing of variables in the stack.

Table 4 lists the SC96AH Series instructions, their opcodes, and execution times. The modification to the flag setting is shown for each instruction. A checkmark (✓) means that the flag is set or cleared as appropriate. A hyphen (-) means that the flag is not modified. A one or zero (1) or (0) indicates that the flag will be in that state after the instruction. An up arrow (↑) indicates that the instruction may set the flag if it is appropriate but will not clear the flag. A down arrow (↓) indicates that the flag can be cleared but not set by the instruction. A question mark (?) indicates that the flag will be left in an indeterminate state after the operation.

Table 4. Instruction Summary

MNEMONIC	OPERANDS	DIRECT			IMMEDIATE			INDIRECT ①					INDEXED ①					OPERATION ③	FLAGS						NOTES
								NORMAL			AUTO-INC.		SHORT			LONG									
		OPCODE	BYTES	STATE TIMES	OPCODE	BYTES	STATE TIMES	OPCODE	BYTES	STATE TIMES⑬	BYTES	STATE TIMES ⑬	OPCODE	BYTES	STATE TIMES ⑬	BYTES	STATE TIMES ⑬		Z	N	C	V	VT	ST	
ARITHMETIC INSTRUCTIONS																									
ADD	2	64	3	4	65	4	5	66	3	6/11	3	7/12	67	4	6/11	5	7/12	D ← D + A	✓	✓	✓	✓	↑	–	
ADD	3	44	4	5	45	5	6	46	4	7/12	4	8/13	47	5	7/12	6	8/13	D ← B + A	✓	✓	✓	✓	↑	–	
ADDB	2	74	3	4	75	3	4	76	3	6/11	3	7/12	77	4	6/11	5	7/12	D ← D + A	✓	✓	✓	✓	↑	–	
ADDB	3	54	4	5	55	4	5	56	4	7/12	4	8/13	57	5	7/12	6	8/13	D ← B + A	✓	✓	✓	✓	↑	–	
ADDC	2	A4	3	4	A5	4	5	A6	3	6/11	3	7/12	A7	4	6/11	5	7/12	D ← D + A + C	↓	✓	✓	✓	↑	–	
ADDCB	2	B4	3	4	B5	3	4	B6	3	6/11	3	7/12	B7	4	6/11	5	7/12	D ← D + A + C	↓	✓	✓	✓	↑	–	
SUB	2	68	3	4	69	4	5	6A	3	6/11	3	7/12	6B	4	6/11	5	7/12	D ← D – A	✓	✓	✓	✓	↑	–	
SUB	3	48	4	5	49	5	6	4A	4	7/12	4	8/13	4B	5	7/12	6	8/13	D ← B – A	✓	✓	✓	✓	↑	–	
SUBB	2	78	3	4	79	3	4	7A	3	6/11	3	7/12	7B	4	6/11	5	7/12	D ← D – A	✓	✓	✓	✓	↑	–	
SUBB	3	58	4	5	59	4	5	5A	4	7/12	4	8/13	5B	5	7/12	6	8/13	D ← B – A	✓	✓	✓	✓	↑	–	
SUBC	2	A8	3	4	A9	4	5	AA	3	6/11	3	7/12	AB	4	6/11	5	7/12	D ← D – A + C – 1	↓	✓	✓	✓	↑	–	
SUBCB	2	B8	3	4	B9	3	4	BA	3	6/11	3	7/12	BB	4	6/11	5	7/12	D ← D – A + C – 1	↓	✓	✓	✓	↑	–	
CMP	2	88	3	4	89	4	5	8A	3	6/11	3	7/12	8B	4	6/11	5	7/12	D – A	✓	✓	✓	✓	↑	–	
CMPB	2	98	3	4	99	3	4	9A	3	6/11	3	7/12	9B	4	6/11	5	7/12	D – A	✓	✓	✓	✓	↑	–	
MULU	2	6C	3	25	6D	4	26	6E	3	27/32	3	28/33	6F	4	27/32	5	28/33	D, D + 2 ← D • A	–	–	–	–	–	?	4
MULU	3	4C	4	26	4D	5	27	4E	4	28/33	4	29/34	4F	5	28/33	6	29/34	D, D + 2 ← B • A	–	–	–	–	–	?	4
MULUB	2	7C	3	17	7D	3	17	7E	3	19/24	3	20/25	7F	4	19/24	5	20/25	D, D + 1 ← D • A	–	–	–	–	–	?	5
MULUB	3	5C	4	18	5D	4	18	5E	4	20/25	4	21/26	5F	5	20/25	6	21/26	D, D + 1 ← B • A	–	–	–	–	–	?	5
MUL	2	②	4	29	②	5	30	②	4	31/36	4	32/37	②	5	31/36	6	32/37	D, D + 2 ← D • A	–	–	–	–	–	?	4
MUL	3	②	5	30	②	6	31	②	5	32/37	5	33/38	②	6	32/37	7	33/38	D, D + 2 ← B • A	–	–	–	–	–	?	4
MULB	2	②	4	21	②	4	21	②	4	23/28	4	24/29	②	5	23/28	6	24/29	D, D + 1 ← D • A	–	–	–	–	–	?	5
MULB	3	②	5	22	②	5	22	②	5	24/29	5	25/30	②	6	24/29	7	25/30	D, D + 1 ← B • A	–	–	–	–	–	?	5
DIVU	2	8C	3	25	8D	4	26	8E	3	28/32	3	29/33	8F	4	28/32	5	29/33	D ← (D, D + 2)/A, D + 2 ← remainder	–	–	–	✓	↑	–	4
DIVUB	2	9C	3	17	9D	3	17	9E	3	20/24	3	21/25	9F	4	20/24	5	21/25	D ← (D, D + 1)/A, D + 1 ← remainder	–	–	–	✓	↑	–	5
DIV	2	②	4	29	②	5	30	②	4	32/36	4	33/37	②	5	32/36	6	33/37	D ← (D, D + 2)/A, D + 2 ← remainder	–	–	–	?	↑	–	
DIVB	2	②	4	21	②	4	21	②	4	24/28	4	25/29	②	5	24/28	6	25/29	D ← (D, D + 1)/A, D + 1 ← remainder	–	–	–	?	↑	–	

Table 4. Instruction Summary (Continued)

MNEMONIC	OPERANDS	DIRECT			IMMEDIATE			INDIRECT ①					INDEXED ①					OPERATION ③	FLAGS						NOTES
								NORMAL			AUTO-INC.		SHORT			LONG									
		OPCODE	BYTES	STATE TIMES	OPCODE	BYTES	STATE TIMES	OPCODE	BYTES	STATE TIMES ⑬	BYTES	STATE TIMES ⑬	OPCODE	BYTES	STATE TIMES ⑬	BYTES	STATE TIMES ⑬		Z	N	C	V	VT	ST	
LOGICAL INSTRUCTIONS																									
AND	2	60	3	4	61	4	5	62	3	6/11	3	7/12	63	4	6/11	5	7/12	D ← D and A	✓	✓	0	0	–	–	
AND	3	40	4	5	41	5	6	42	4	7/12	4	8/13	43	5	7/12	6	8/13	D ← B and A	✓	✓	0	0	–	–	
ANDB	2	70	3	4	71	3	4	72	3	6/11	3	7/12	73	4	6/11	5	7/12	D ← D and A	✓	✓	0	0	–	–	
ANDB	3	50	4	5	51	4	5	52	4	7/12	4	8/13	53	5	7/12	6	8/13	D ← B and A	✓	✓	0	0	–	–	
OR	2	80	3	4	81	4	5	82	3	6/11	3	7/12	83	4	6/11	5	7/12	D ← D or A	✓	✓	0	0	–	–	
ORB	2	90	3	4	91	3	4	92	3	6/11	3	7/12	93	4	6/11	5	7/12	D ← D or A	✓	✓	0	0	–	–	
XOR	2	84	3	4	85	4	5	86	3	6/11	3	7/12	87	4	6/11	5	7/12	D ← D (excl. or) A	✓	✓	0	0	–	–	
XORB	2	94	3	4	95	3	4	96	3	6/11	3	7/12	97	4	6/11	5	7/12	D ← D (excl. or) A	✓	✓	0	0	–	–	
DATA TRANSFER INSTRUCTIONS																									
LD	2	A0	3	4	A1	4	5	A2	3	6/11	3	7/12	A3	4	6/11	5	7/12	D ← A	–	–	–	–	–	–	
LDB	2	B0	3	4	B1	3	4	B2	3	6/11	3	7/12	B3	4	6/11	5	7/12	D ← A	–	–	–	–	–	–	
ST	2	C0	3	4	–	–	–	C2	3	7/11	3	8/12	C3	4	7/11	5	8/12	A ← D	–	–	–	–	–	–	
STB	2	C4	3	4	–	–	–	C6	3	7/11	3	8/12	C7	4	7/11	5	8/12	A ← D	–	–	–	–	–	–	
LDBSE	2	BC	3	4	BD	3	4	BE	3	6/11	3	7/12	BF	4	6/11	5	7/12	D ← A; D + 1 ← SIGN(A)	–	–	–	–	–	–	5,6
LDBZE	2	AC	3	4	AD	3	4	AE	3	6/11	3	7/12	AF	4	6/11	5	7/12	D ← A; D + 1 ← 0	–	–	–	–	–	–	5,6
STACK OPERATIONS (Internal stack)																									
PUSH	1	C8	2	8	C9	3	8	CA	2	11/15	2	12/16	CB	3	11/15	4	12/16	SP ← SP – 2; (SP) ← A	–	–	–	–	–	–	
POP	1	CC	2	12	–	–	–	CE	2	14/18	2	14/18	CF	3	14/18	4	14/18	A ← (SP); SP ← SP + 2	–	–	–	–	–	–	
PUSHF	0	F2	1	8														SP ← SP – 2; (SP) ← PSW; PSW ← 0000H I ← 0	0	0	0	0	0	0	
POPF	0	F3	1	9														PSW ← (SP); SP ← SP + 2 I ← ✓	✓	✓	✓	✓	✓	✓	
STACK OPERATIONS (external stack)																									
PUSH	1	C8	2	12	C9	3	12	CA	2	15/19	2	16/20	CB	3	15/19	4	16/20	SP ← SP – 2; (SP) ← A	–	–	–	–	–	–	
POP	1	CC	2	14	–	–	–	CE	2	16/20	2	16/20	CF	3	16/20	4	16/20	A ← (SP); SP ← SP + 2	–	–	–	–	–	–	
PUSHF	0	F2	1	12														SP ← SP – 2; (SP) ← PSW; PSW ← 0000H I ← 0	0	0	0	0	0	0	
POPF	0	F3	1	13														PSW ← (SP); SP ← SP + 2 I ← ✓	✓	✓	✓	✓	✓	✓	

Table 4. Instruction Summary (Continued)

JUMPS AND CALLS

MNEMONIC	OPCODE	BYTES	STATES	OPERATION ③	FLAGS						NOTES
					Z	N	C	V	VT	ST	
LJMP	E7	3	8	PC ← PC + 16-bit offset	–	–	–	–	–	–	10
SJMP	20 – 27	2	8	PC ← PC + 11-bit offset	–	–	–	–	–	–	10, 15
BR[ ]	E3	2	8	PC ← (A)	–	–	–	–	–	–	
LCALL	EF	3	13/16	SP ← SP – 2; (SP) ← PC; PC ← PC + 16-bit offset	–	–	–	–	–	–	10, 14
SCALL	28 – 2F	2	13/16	SP ← SP – 2; (SP) ← PC; PC ← PC + 11-bit offset	–	–	–	–	–	–	10, 14, 15
RET	F0	1	12/16	PC ← (SP); SP ← SP + 2	–	–	–	–	–	–	14
TRAP	F7	1	21/24	SP ← SP – 2; (SP) ← PC PC ← (2010H)	–	–	–	–	–	–	9, 14

CONDITIONAL JUMPS

All conditional jumps are 2 byte instructions. They require 8 state times if the jump is taken, 4 if it is not.
PC ← PC + 8-bit offset (if taken)

MNEMONIC	OPCODE	OPERATION ③	FLAGS						NOTES
			Z	N	C	V	VT	ST	
JC	DB	Jump if C = 1	–	–	–	–	–	–	10
JNC	D3	Jump if C = 0	–	–	–	–	–	–	10
JE	DF	Jump if Z = 1	–	–	–	–	–	–	10
JNE	D7	Jump if Z = 0	–	–	–	–	–	–	10
JGE	D6	Jump if N = 0	–	–	–	–	–	–	10
JLT	DE	Jump if N = 1	–	–	–	–	–	–	10
JGT	D2	Jump if N = 0 and Z = 0	–	–	–	–	–	–	10
JLE	DA	Jump if N = 1 or Z = 1	–	–	–	–	–	–	10
JH	D9	Jump if C = 1 and Z = 0	–	–	–	–	–	–	10
JNH	D1	Jump if C = 0 or Z = 1	–	–	–	–	–	–	10
JV	DD	Jump if V = 1	–	–	–	–	–	–	10
JNV	D5	Jump if V = 0	–	–	–	–	–	–	10
JVT	DC	Jump if VT = 1; Clear VT	–	–	–	–	0	–	10
JNVT	D4	Jump if VT = 0; Clear VT	–	–	–	–	0	–	10
JST	D8	Jump if ST = 1	–	–	–	–	–	–	10
JNST	D0	Jump if ST = 0	–	–	–	–	–	–	10

JUMP ON BIT CLEAR OR BIT SET/LOOP CONTROL

These instructions are 3-byte instructions. They require 9 state times if the jump is taken, 5 if it is not.

MNEMONIC	OPCODE	OPERATION ③	Z	N	C	V	VT	ST	NOTES
JBS	38-3F	Jump if Specified Bit = 1	–	–	–	–	–	–	10, 11
JBC	30-37	Jump if Specified Bit = 0	–	–	–	–	–	–	10, 11
DJNZ	E0	D ← D – 1; if D ≠ 0 then PC ← PC + 8-bit offset	–	–	–	–	–	–	10

Table 4. Instruction Summary (Continued)

SINGLE REGISTER INSTRUCTIONS

These instructions are 2-byte instructions. They require 4 state times.

MNEMONIC	OPCODE	OPERATION 3	FLAGS						NOTES
			Z	N	C	V	VT	ST	
DEC	05	D ← D − 1	✓	✓	✓	✓	↑	–	
DECB	15	D ← D − 1	✓	✓	✓	✓	↑	–	
NEG	03	D ← 0 − D	✓	✓	✓	✓	↑	–	
NEGB	13	D ← 0 − D	✓	✓	✓	✓	↑	–	
INC	07	D ← D + 1	✓	✓	✓	✓	↑	–	
INCB	17	D ← D + 1	✓	✓	✓	✓	↑	–	
EXT	06	D ← D; D + 2 ← Sign (D)	✓	✓	0	0	–	–	4
EXTB	16	D ← D; D + 1 ← Sign (D)	✓	✓	0	0	–	–	5
NOT	02	D ← Logical Not (D)	✓	✓	0	0	–	–	
NOTB	12	D ← Logical Not (D)	✓	✓	0	0	–	–	
CLR	01	D ← 0	1	0	0	0	–	–	
CLRB	11	D ← 0	1	0	0	0	–	–	

SHIFT INSTRUCTIONS

These instructions are 3-byte instructions

MNEMONIC	OPCODE	STATE TIMES	OPERATION ③	FLAGS						NOTES
				Z	N	C	V	VT	ST	
NORML	0F	11 + 1 PER SHIFT	Left shift till msb = 1; D ← shift count	✓	?	0	–	–	–	12
SHL	09	7 + 1 PER SHIFT	C ← msb − − − − − lsb ← 0	✓	?	✓	✓	↑	–	12
SHR	08	7 + 1 PER SHIFT	0 → msb − − − − − lsb → C	✓	?	✓	0	–	✓	12
SHRA	0A	7 + 1 PER SHIFT	msb → msb − − − − − lsb → C	✓	✓	✓	0	–	✓	12
SHLB	19	7 + 1 PER SHIFT	C ← msb − − − − − lsb ← 0	✓	?	✓	✓	↑	–	12
SHRB	18	7 + 1 PER SHIFT	0 → msb − − − − − lsb → C	✓	?	✓	0	–	✓	12
SHRAB	1A	7 + 1 PER SHIFT	msb → msb − − − − − lsb → C	✓	✓	✓	0	–	✓	12
SHLL	0D	7 + 1 PER SHIFT	C ← msb − − − − − lsb ← 0	✓	?	✓	✓	↑	–	7, 12
SHRL	0C	7 + 1 PER SHIFT	0 → msb − − − − − lsb → C	✓	?	✓	0	–	✓	7, 12
SHRAL	0E	7 + 1 PER SHIFT	msb → msb − − − − − lsb → C	✓	✓	✓	0	–	✓	7, 12

SPECIAL CONTROL INSTRUCTIONS

These instructions are 1-byte instructions except SKIP is 2-bytes.

MNEMONIC	OPCODE	STATE TIMES	OPERATION	Z	N	C	V	VT	ST	NOTES
SETC	F9	4	C ← 1	–	–	1	–	–	–	
CLRC	F8	4	C ← 0	–	–	0	–	–	–	
CLRVT	FC	4	VT ← 0	–	–	–	–	0	–	
RST	FF	16	PC ← 2080H	0	0	0	0	0	0	8
DI	FA	4	Disable All Interrupts (I ← 0)	–	–	–	–	–	–	
EI	FB	4	Enable All Interrupts (I ← 1)	–	–	–	–	–	–	
NOP	FD	4	PC ← PC + 1	–	–	–	–	–	–	
SKIP	00	4	PC ← PC + 2	–	–	–	–	–	–	

NOTES:

1. Long Indexed and Indirect + instructions have identical opcodes with Short Indexed and Indirect modes, respectively. The second byte of instructions using any indirect or indexed addressing mode specifies the exact mode used. If the second byte is even, use Indirect or Short Indexed. If it is odd, use Indirect + or Long Indexed. In all cases the second byte of the instruction always specifies an even (word) location for the address referenced.
2. The opcodes for signed multiply and divide are the opcodes for the unsigned functions with an "FE" appended as a prefix.
3. If the mnemonic ends in "B", a byte operation is performed, otherwise a word operation is done. Operands D, B, and A must conform to the alignment rules for the required operand type. D and B are locations in the register file; A can be located anywhere in memory.
4. D, D + 2 are consecutive WORDS in memory; D is DOUBLE-WORD aligned.
5. D, D + 1 are consecutive BYTES in memory; D is WORD aligned.
6. Changes a byte to a word.
7. The "L" (Long) suffix indicates double-word operation.
8. Initiates a reset by pulling $\overline{\text{RESET}}$ low. Software should re-initialize all the necessary registers with code starting at 2080H.
9. The assembler will not accept this mnemonic.
10. Offset is a 2's complement number.
11. Specified bit is one of the 2048 bits in the register file.
12. Execution will take at least 8 states, even for 0 shift.
13. Number of state times shown for internal/external operands.
14. State times for stack located internal/external.
15. The least significant 3 bits of the opcode are concatenated with the following 8 bits to form an 11-bit, 2's complement, offset for the relative call or jump.

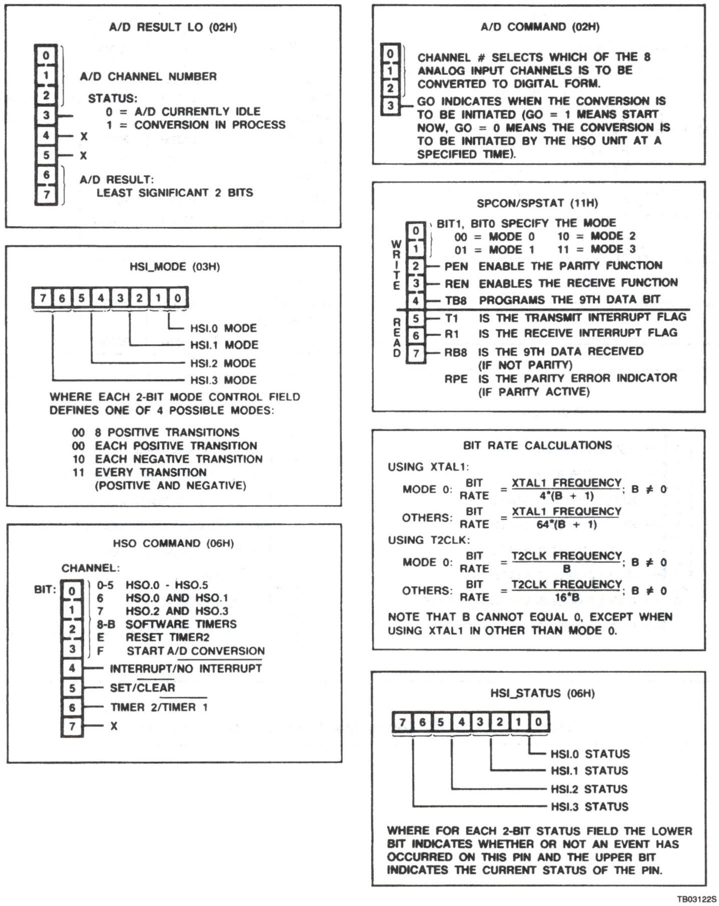

A/D RESULT LO (02H)
0 1 2 A/D CHANNEL NUMBER
3 STATUS:
0 = A/D CURRENTLY IDLE
1 = CONVERSION IN PROCESS
4 X
5 X
6 7 A/D RESULT:
LEAST SIGNIFICANT 2 BITS
A/D COMMAND (02H)
0 1 2 CHANNEL # SELECTS WHICH OF THE 8 ANALOG INPUT CHANNELS IS TO BE CONVERTED TO DIGITAL FORM.
3 GO INDICATES WHEN THE CONVERSION IS TO BE INITIATED (GO = 1 MEANS START NOW, GO = 0 MEANS THE CONVERSION IS TO BE INITIATED BY THE HSO UNIT AT A SPECIFIED TIME).
HSI_MODE (03H)
7 6 5 4 3 2 1 0
HSI.0 MODE
HSI.1 MODE
HSI.2 MODE
HSI.3 MODE
WHERE EACH 2-BIT MODE CONTROL FIELD DEFINES ONE OF 4 POSSIBLE MODES:
00 8 POSITIVE TRANSITIONS
00 EACH POSITIVE TRANSITION
10 EACH NEGATIVE TRANSITION
11 EVERY TRANSITION (POSITIVE AND NEGATIVE)
SPCON/SPSTAT (11H)
WRITE
READ
0 1 BIT1, BIT0 SPECIFY THE MODE
00 = MODE 0 10 = MODE 2
01 = MODE 1 11 = MODE 3
2 PEN ENABLE THE PARITY FUNCTION
3 REN ENABLES THE RECEIVE FUNCTION
4 TB8 PROGRAMS THE 9TH DATA BIT
5 T1 IS THE TRANSMIT INTERRUPT FLAG
6 R1 IS THE RECEIVE INTERRUPT FLAG
7 RB8 IS THE 9TH DATA RECEIVED (IF NOT PARITY)
RPE IS THE PARITY ERROR INDICATOR (IF PARITY ACTIVE)
BIT RATE CALCULATIONS
USING XTAL1:
MODE 0: BIT RATE = XTAL1 FREQUENCY / 4*(B + 1); B ≠ 0
OTHERS: BIT RATE = XTAL1 FREQUENCY / 64*(B + 1)
USING T2CLK:
MODE 0: BIT RATE = T2CLK FREQUENCY / B; B ≠ 0
OTHERS: BIT RATE = T2CLK FREQUENCY / 16*B; B ≠ 0
NOTE THAT B CANNOT EQUAL 0, EXCEPT WHEN USING XTAL1 IN OTHER THAN MODE 0.
HSO COMMAND (06H)
CHANNEL:
BIT: 0 1 2 3
0-5 HSO.0 - HSO.5
6 HSO.0 AND HSO.1
7 HSO.2 AND HSO.3
8-B SOFTWARE TIMERS
E RESET TIMER2
F START A/D CONVERSION
4 INTERRUPT/$\overline{\text{NO INTERRUPT}}$
5 SET/$\overline{\text{CLEAR}}$
6 TIMER 2/$\overline{\text{TIMER 1}}$
7 X
HSI_STATUS (06H)
7 6 5 4 3 2 1 0
HSI.0 STATUS
HSI.1 STATUS
HSI.2 STATUS
HSI.3 STATUS
WHERE FOR EACH 2-BIT STATUS FIELD THE LOWER BIT INDICATES WHETHER OR NOT AN EVENT HAS OCCURRED ON THIS PIN AND THE UPPER BIT INDICATES THE CURRENT STATUS OF THE PIN.
TB03122S

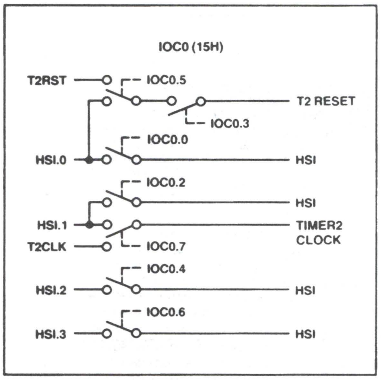
Figure 14. SC96AH Series Quick Reference

IOC0 (15H)

0 — HSI.0 INPUT ENABLE/$\overline{\text{DISABLE}}$
1 — TIMER 2 RESET EACH WRITE
2 — HSI.1 INPUT ENABLE/$\overline{\text{DISABLE}}$
3 — TIMER 2 EXTERNAL RESET ENABLE/$\overline{\text{DISABLE}}$
4 — HSI.2 INPUT ENABLE/$\overline{\text{DISABLE}}$
5 — TIMER 2 RESET SOURCE HSI.0/$\overline{\text{T2RST}}$
6 — HSI.3 INPUT ENABLE/$\overline{\text{DISABLE}}$
7 — TIMER 2 CLOCK SOURCE HSI.1/$\overline{\text{T2CLK}}$

IOC1 (16H)

0 — SELECT PWM/$\overline{\text{SELECT P2.5}}$
1 — EXTERNAL INTERRUPT ACH7/$\overline{\text{EXTINT}}$
2 — TIMER 1 OVERFLOW INTERRUPT ENABLE/$\overline{\text{DISABLE}}$
3 — TIMER 2 OVERFLOW INTERRUPT ENABLE/$\overline{\text{DISABLE}}$
4 — HSO.4 OUTPUT ENABLE/$\overline{\text{DISABLE}}$
5 — SELECT TXD/$\overline{\text{SELECT P2.0}}$
6 — HSO.5 OUTPUT ENABLE/$\overline{\text{DISABLE}}$
7 — HSI INTERRUPT FIFO FULL/$\overline{\text{HOLDING REGISTER LOADED}}$



VECTOR	VECTOR LOCATION		PRIORITY
	(HIGH BYTE)	(LOW BYTE)	
SOFTWARE	2011H	2010H	NOT APPLICABLE
EXTINT	200FH	200EH	7 (HIGHEST)
SERIAL PORT	200DH	200CH	6
SOFTWARE TIMERS	200BH	200AH	5
HSI.0	2009H	2008H	4
HIGH SPEED OUTPUTS	2007H	2006H	3
HSI DATA AVAILABLE	2005H	2004H	2
A/D CONVERSION COMPLETE	2003H	2002H	1
TIMER OVERFLOW	2001H	2000H	0 (LOWEST)

IOS0 (15H)

0 — HSO.0 CURRENT STATE
1 — HSO.1 CURRENT STATE
2 — HSO.2 CURRENT STATE
3 — HSO.3 CURRENT STATE
4 — HSO.4 CURRENT STATE
5 — HSO.5 CURRENT STATE
6 — CAM OR HOLDING REGISTER IS FULL
7 — HSO HOLDING REGISTER IS FULL

IOS1 (16H)

0 — SOFTWARE TIMER 0 EXPIRED
1 — SOFTWARE TIMER 1 EXPIRED
2 — SOFTWARE TIMER 2 EXPIRED
3 — SOFTWARE TIMER 3 EXPIRED
4 — TIMER 2 HAS OVERFLOW
5 — TIMER 1 HAS OVERFLOW
6 — HSI FIFO IS FULL
7 — HSI HOLDING REGISTER DATA AVAILABLE

TB03130S

Figure 14. SC96AH Series Quick Reference (Continued)

ABSOLUTE MAXIMUM RATINGS [1]

SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	0 to +70	°C
T_{STG}	Storage temperature	−40 to +150	°C
	Voltage from any pin to V_{SS} or ANGND[3]	−0.3 to +7.0	V
I_{OUT}	Average output current from any pin	10	mA
P_D	Power dissipation	1.5	W

OPERATING CONDITIONS [4, 5, 6]

SYMBOL	PARAMETER	MIN	MAX	UNIT
T_A	Ambient temperature	0	+70	°C
V_{CC}	Digital supply voltage	4.50	5.50	V
V_{REF}	Analog supply voltage	4.50 $V_{CC} - 0.3$	5.50 $V_{CC} + 0.3$	V
f_{OSC}	Oscillator frequency	6.0	12	MHz
V_{PD}	Power-down supply voltage	4.50	5.50	V

DC ELECTRICAL CHARACTERISTICS [4, 5]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			Min	Max	
V_{IL}	Input low voltage (except $\overline{\text{RESET}}$)		−0.3	+0.8	V
V_{IL1}	Input low voltage, $\overline{\text{RESET}}$		−0.3	+0.7	V
V_{IH}	Input high voltage (except $\overline{\text{RESET}}$, NMI, XTAL1)		2.0	$V_{CC} + 0.5$	V
V_{IH1}	Input high voltage, $\overline{\text{RESET}}$ rising		2.4	$V_{CC} + 0.5$	V
V_{IH2}	Input high voltage, $\overline{\text{RESET}}$ falling		2.1	$V_{CC} + 0.5$	V
V_{IH3}	Input high voltage, NMI, XTAL1		2.2	$V_{CC} + 0.5$	V
V_{OL}	Output low voltage	(Note 7)		0.45	V
V_{OH}	Output high voltage	(Note 8)	2.4		V
I_{CC}	V_{CC} supply current	All outputs disconnected.		200	mA
I_{PD}	V_{PD} supply current	Normal operation and power-down		1	mA
I_{REF}	V_{REF} supply current			8	mA
I_{LI}	Input leakage current to all pins of HSI, P0, P3, P4, and to P2.1.	$V_{in} = 0$ to V_{CC} (Note 9)		±10	μA
I_{IH}	Input high current to $\overline{\text{EA}}$	$V_{IH} = 2.4V$		100	μA
I_{IL}	Input low current to all pins of P1, and to P2.6, P2.7.	$V_{IL} = 0.45V$		−100	μA
I_{IL1}	Input low current to $\overline{\text{RESET}}$	$V_{IL} = 0.45V$		−2	mA
I_{IL2}	Input low current P2.2, P2.3, P2.4, READY	$V_{IL} = 0.45V$		−50	μA
C_S	Pin capacitance (any pin to V_{SS})	fTEST = 1.0MHz		10	pF

A/D CONVERTER CHARACTERISTICS [4, 10]

PARAMETER	RATING	UNIT
Resolution	±0.001	V_{REF}
Accuracy	±0.004	V_{REF}
Differential nonlinearity (max)	±0.002	V_{REF}
Integral nonlinearity (max)	±0.004	V_{REF}
Channel-to-channel matching	±1	LSB
Crosstalk, DC to 100KHz (max)	−60	dB

AC ELECTRICAL CHARACTERISTICS [4, 5, 11]

Timing Requirements (other system components must meet these specifications)

SYMBOL	PARAMETER	MIN	MAX	UNIT
t_{CLYX}[16]	READY hold after CLKOUT edge[12]	0		ns
t_{LLYV}	End of ALE to READY setup	$-T_{OSC}$	$2T_{OSC}-60$	ns
t_{LLYH}	End of ALE to READY high[13]	$2T_{OSC}+60$	$4T_{OSC}-60$	ns
t_{YLYH}	Non-ready time		1000	ns
t_{AVDV}	Address valid to input data valid		$5T_{OSC}-90$	ns
t_{RLDV}	$\overline{\text{RD}}$ active to input data valid		$3T_{OSC}-60$	ns
t_{RHDX}	Data hold after $\overline{\text{RD}}$ inactive[14]	0		ns
t_{RHDZ}	$\overline{\text{RD}}$ inactive to input data float[14]	0	$T_{OSC}-20$	ns

Timing Responses (SC96AH parts meet these specifications)

SYMBOL	PARAMETER	TENTATIVE LIMITS		UNIT
		Min	Max	
F_{XTAL}	Oscillator frequency	6	12	MHz
T_{OSC}	Oscillator period	83	166	ns
T_{CHCH}[16]	CLKOUT period[14, 15]	$3T_{OSC}$	$3T_{OSC}$	ns
t_{CHCL}[16]	CLKOUT High time	$T_{OSC}-20$	$T_{OSC}+20$	ns
t_{CLLH}[16]	CLKOUT low to ALE High	-5	20	ns
t_{LLCH}[16]	ALE low to CLKOUT High	$T_{OSC}-20$	$T_{OSC}+40$	ns
t_{LHLL}	ALE pulse width	$T_{OSC}-25$	$T_{OSC}+15$	ns
t_{AVLL}	Address setup to end of ALE	$T_{OSC}-50$		ns
t_{LLRL}	End of ALE to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ active	$T_{OSC}-20$		ns
t_{LLAX}	Address hold after end of ALE	$T_{OSC}-20$		ns
t_{WLWH}	$\overline{\text{WR}}$ pulse width	$2T_{OSC}-35$		ns
t_{QVWH}	Output data valid to end of $\overline{\text{WR}}$	$2T_{OSC}-60$		ns
t_{WHQX}	Output data hold after $\overline{\text{WR}}$	$T_{OSC}-25$		ns
t_{WLLH}	End of $\overline{\text{WR}}$ to next ALE	$T_{OSC}-30$		ns
t_{RLRH}	$\overline{\text{RD}}$ pulse width	$3T_{OSC}-30$		ns
t_{RHLH}	End of $\overline{\text{RD}}$ to next ALE	$T_{OSC}-25$		ns

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions other than those described in the Operating Conditions section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over Operating Conditions unless otherwise specified.
5. All voltage measurements are referenced to ground. For testing, all input signals swing between 0.45V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2V and at output voltages of 0.8V and 2V as appropriate.
6. ANGND and V_{SS} should be nominally at the same potential.
7. I_{OL} = 0.36mA for all pins of P1, for P2.6 and P2.7, and for all pins of P3 and P4 when used as ports. I_{OL} = 2mA for TXD, RXD (in serial port mode 0), PWM, CLKOUT, ALE, $\overline{\text{BHE}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, and all pins of HSO and P3 and P4 when used as external memory bus (AD0 – AD15).
8. $I_{OH} = -20\mu A$ for all pins of P1, P2.6 and P2.7. $I_{OH} = -200\mu A$ for TXD, RXD (in serial port mode 0), PWM, CLKOUT, ALE, $\overline{\text{BHE}}$, $\overline{\text{WR}}$, and all pins of HSO and P3 and P4 when used as external memory bus (AD0 – AD15). P3 and P4, when used as ports, have open-drain outputs.
9. Analog conversion not in process.
10. A/D converter operation is verified only on the 8097AH, 8397AH, 8095AH and 8395AH. The absolute conversion accuracy is dependent on the accuracy of V_{REF}. Testing is done at V_{REF} = 5.120V.
11. Test conditions for AC characteristics: load capacitance on output pins = 80pF; oscillator frequency = 12MHz.
12. If the 48-pin part is being used then this timing can be generated by assuming that the CLKOUT falling edge has occurred at $2T_{OSC+60}$ (t_{LLCH}(max) + t_{CHCL}(max)) after the falling edge of ALE.
13. If more than one wait state is desired, add $3T_{OSC}$ for each additional wait state.
14. This specification is not tested, but is verified by design analysis and/or derived from other tested parameters.
15. CLKOUT is directly generated as a divide by three of the oscillator. The period will be $3T_{OSC}$ ± 10ns if T_{OSC} is constant and the rise and fall times on XTAL1 are less than 10ns.
16. Pins not bonded out on 48-pin parts.

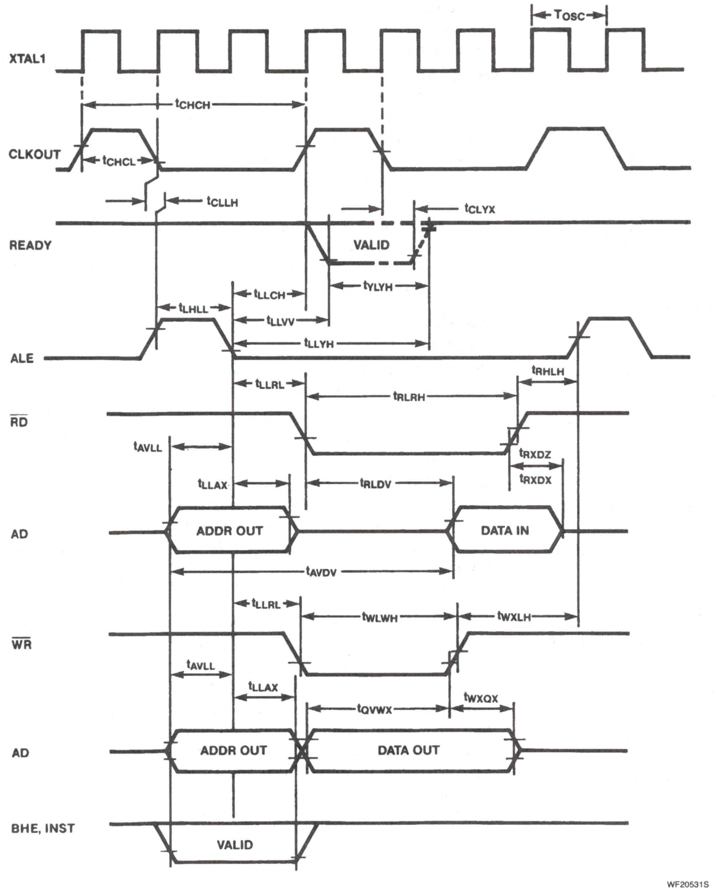

T_{OSC}
XTAL1
t_{CHCH}
CLKOUT
t_{CHCL}
t_{CLLH}
t_{CLYX}
READY
VALID
t_{LLCH}
t_{YLYH}
t_{LHLL}
t_{LLVV}
t_{LLYH}
ALE
t_{RHLH}
t_{LLRL}
t_{RLRH}
$\overline{RD}$
t_{AVLL}
t_{RXDZ}
t_{LLAX}
t_{RLDV}
t_{RXDX}
AD
ADDR OUT
DATA IN
t_{AVDV}
t_{LLRL}
t_{WLWH}
t_{WXLH}
$\overline{WR}$
t_{AVLL}
t_{LLAX}
t_{WXQX}
t_{QVWX}
AD
ADDR OUT
DATA OUT
BHE, INST
VALID
WF20531S

Figure 15. Bus Signal Timing

Single-Chip 16-Bit Microcontrollers

SC96AH Series

FUNCTIONAL DEVIATIONS

Signetics SC96AH series devices are functionally identical to Intel 809X-90 and 839X-90 devices. Functional deviations are as listed below.

CPU Section

1. Indexed, 3 Word Multiply — The displacement portion of an indexed, three word multiply may not be in the range of 200H thru 17FFH inclusive.
2. Add or Substract with carry — The zero flag is both set and cleared by these instructions.
3. EXT — This instruction never sets the N flag, and always sets the Z flag. EXTB works correctly.
4. Read-Modify-Write on Interrupt Pending — A read-modify-write instruction on the interrupt pending register may cause interrupts that occur during execution of the instruction to be missed.
5. READY line — The READY line should not be held active during the execution of an instruction that accesses HSI_TIME, SP_STAT or IOS1. It should also not be active for a data write during the instruction immediately preceding one of the above operations.
6. Signed Divide — The V and VT flags may indicate an overflow after a signed divide when no overflow has occurred.

HSI/HSO Section

1. HSI Timing — An event occurring within 16 state times of a prior event on the same HSI line may not be recorded. Additionally, an event occurring within 16 state times of a prior event on another HSI line may be recorded with a time tag one count earlier than expected. Events are defined as the condition the line is set to trigger on.
2. HSI Divide by 8 Mode — If an event on a pin set to look for every eight transition occurs less than 16 state times after an event on any other pin, then the divide by 8 event will be recorded twice in the HSI FIFO. The time tag of the initial entry plus one.
3. HSO Interrupts — Software timer interrupts cannot be generated by the HSO commands that reset Timer 2 or start an A to D conversion.

Serial Port Section

1. Serial Port Flags — Reading SP_STAT may not clear the TI or RI flag was set within two state times prior to the read.
2. Serial Port Mode 0 — The serial port is not tested in mode 0. The receive function in this mode does not work correctly.
3. Serial Port Baud Value — Loading the baud rate register with 8000H (maximum baud rate, internal clock) may cause an 11 millisecond delay (at Fosc = 12MHz) before the port is properly initialized. After initialization the port works properly.

Standard I/O Section

1. Ports 3 and 4 (Internal Execution Mode Only) — To be used as outputs, Ports 3 and 4 each must be addressed as words but written to as bytes. To write to Port 3 use "ST temp, 1ffeh", where the low byte of "temp" contains the data for the port. To write to Port 4, use the DCB operator to generate the opcode sequence "0C3H, 001H, 0FFH, 01FH, (temp)", where the high byte of "temp" contains the data for the port. Ports 3 and 4 will not work as input ports.

Additional Information

1. After the chip reset the watchdog timer will not run until a "01EH" followed by a "0E1H" is written to the watchdog timer register. This feature permits disabling of the watchdog by simply not writing to it.
2. External interrupts on P0.7 are sampled every state time instead of every eight state times.
3. If more than one HSO event is scheduled to occur with interrupts at the same time multiple HSO interrupts may occur. This is because HSO interrupts are internal events and as such are not synchronized to timer1.
4. Locations 2012H through 207FH in external memory must be filled with the hex value 0FFFFH to ensure compatibility with future parts. The internal locations in this range are still reserved for the factory test code.
5. There are several restrictions on using the special function registers.
   - A. Neither the source not the destination addresses of the multiply and divide instructions can be a writable special function register.
   - B. These registers may not be used as base or index registers for indexed or indirect instructions.
   - C. Several of these registers can only be accessed as words, while others only as bytes. These restrictions are listed in Table 1.

SC96BH Series
Single-Chip 16-Bit Microcontrollers

Preliminary Specification (Brief)

Microprocessor Products

DESCRIPTION

The SC96BH Series microcontrollers consist of a powerful 16-bit CPU tightly coupled with 8K bytes of program memory, 232 bytes of data memory, and I/O ports all integrated onto a single piece of silicon. This combination of resources on a single chip represents the highest level of integration in the world of microcontrollers, making possible new applications which required multiple chip solutions in the past.

The CPU features a register-to-register architecture and a dual-operand ALU which supports bit, byte and word operations, as well as 32-bit double words for a subset of the instructions. With a 12MHz clock frequency, the 8096BH (the generic part number used for SC96BH Series microcontrollers) can perform a 16-bit addition in 1μs and a 16 × 16 multiply or 32/16 divide in 6.25μs. Instruction execution times average 1 to 2μs in typical applications.

Four high-speed trigger inputs can be used to record the times at which external events occur with a resolution of 2μs. Up to six high speed pulse generator outputs are provided to trigger external events at preset times. These high speed inputs and outputs supplement five multi-function 8-bit I/O ports. The high speed output unit can simultaneously perform software timer functions. Up to four such 16-bit timers can be in operation at once in addition to the two 16-bit hardware timers which are on the chip.

Also provided on-chip are a pulse-width modulated output signal, a watchdog timer and a serial port with a dedicated bit-rate generator. Several family members contain an analog-to-digital converter, including a sample-and-hold circuit, which converts up to eight analog inputs into 10-bit digital values. Each conversion takes 22μs with a 12MHz clock.

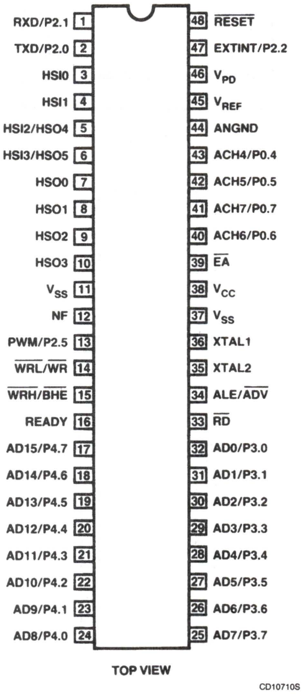
Although 8096BH is the generic part number used throughout this publication, the product family consists of eight part numbers including the 8096BH. This wide variety of products is offered to meet requirements of various users in terms of I/Os and package size. The options include on-board mask programmed memory, 10-bit A/D converter and 48-, 64- or 68-pin packages as shown in the SC96BH Family Table.

The 8096BH can address up to 64K bytes of memory, so that external memory can be used either as the sole program memory (809X versions) or to expand the 8K bytes of internal program ROM (839XBH versions). External memory can also be used for data storage. The 8096BH allows the external memory bus to be configured to operate either as a 16-bit multiplexed address/data bus or as an 8088 CPU minimum mode type bus.

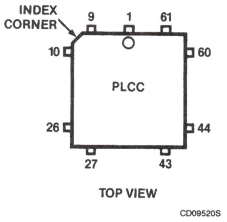
FEATURES

- **Single-chip 16-bit microcontroller**
- **Register to register architecture**
- **8K bytes internal ROM (839XBH versions)**
- **64K byte addressing range**
  - **8- or 16-bit external memory bus width**
- **232 byte register file**
  - **16 bytes power-down RAM available**
- **Versatile 2-operand ALU**
  - **Bit, byte, word and double word operands**
  - **Hardware multiply and divide**
  - **1μs basic instruction cycle**
- **10-bit A/D converter with sample and hold**
  - **22μs conversion time**
- **Pulse width modulated D/A output**
- **Up to five 8-bit I/O ports**
- **High-speed I/O system**
  - **Four inputs and six outputs**
- **Full duplex serial port**
  - **Dedicated bit rate generator**
- **20 interrupt sources**
- **Two 16-bit counter/timers**
- **16-bit watchdog timer**
- **Four 16-bit software timers**
- **ROM security**
- **48-, 64- and 68-pin packages**

PIN CONFIGURATIONS



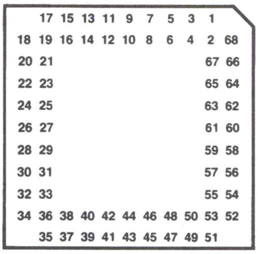
RXD/P2.1 1
TXD/P2.0 2
HSI0 3
HSI1 4
HSI2/HSO4 5
HSI3/HSO5 6
HSO0 7
HSO1 8
HSO2 9
HSO3 10
V_SS 11
NF 12
PWM/P2.5 13
WRL/WR 14
WRH/BHE 15
READY 16
AD15/P4.7 17
AD14/P4.6 18
AD13/P4.5 19
AD12/P4.4 20
AD11/P4.3 21
AD10/P4.2 22
AD9/P4.1 23
AD8/P4.0 24
48 RESET
47 EXTINT/P2.2
46 V_PD
45 V_REF
44 ANGND
43 ACH4/P0.4
42 ACH5/P0.5
41 ACH7/P0.7
40 ACH6/P0.6
39 EA
38 V_CC
37 V_SS
36 XTAL1
35 XTAL2
34 ALE/ADV
33 RD
32 AD0/P3.0
31 AD1/P3.1
30 AD2/P3.2
29 AD3/P3.3
28 AD4/P3.4
27 AD5/P3.5
26 AD6/P3.6
25 AD7/P3.7
TOP VIEW
CD10710S



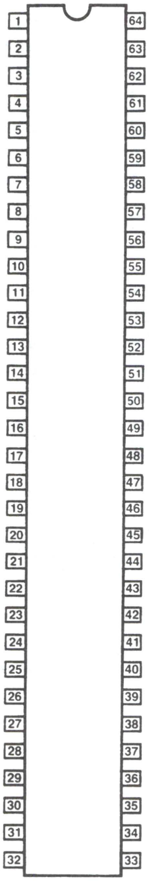
INDEX CORNER
9
1
61
10
60
PLCC
26
44
27
43
TOP VIEW
CD09520S

Single-Chip 16-Bit Microcontrollers SC96BH Series

PIN CONFIGURATIONS



TOP VIEW



NOTE:
See SC96BH Series Data Sheet (now in prep)

FUNCTIONAL DESCRIPTION

As shown in the block diagram, the major components of the 8096BH are a CPU, a programmable high-speed I/O unit, an analog to digital converter, serial and parallel I/O ports, a watchdog timer, and a pulse width modulated output controller. Several additional blocks support these units and overall operation of the chip.

CPU Architecture

The major components of the CPU are the register file and the register/arithmetic logic unit (RALU).

The 8096BH uses the same address space for both program and data memory, except in the address range from 00H to 0FFH. Data fetches in this range are always from the register file, while instruction fetches from these locations are directed to external memory and are reserved for development support tools. Program memory, data memory and peripherals may be intermixed outside of the register file.

Within the register file, locations 00H through 17H are register mapped I/O control registers, also referred to as special function registers (SFRs). The rest of the register file (018H to 0FFH) contains 232 bytes of RAM which can be referenced as bytes, words, or double-words. This register space allows the most frequently used variables to be kept in on-chip RAM, which can be accessed faster than external memory. Locations 0F0H through 0FFH can be preserved during power-down via a separate power pin.

The 809XBH devices operate from external memory only. The 839XBH devices contain 8K bytes of mask programmed ROM for program memory which occupies addresses 2000H through 3FFFH. Instruction or data fetches from these addresses access the on-chip memory if the $\overline{EA}$ pin is held High. If the $\overline{EA}$ pin is held low, these addresses access external memory.

The RALU section consists of a 17-bit ALU, the program status word, the program counter and several temporary registers. A key feature of the 8096BH is that it does not use an accumulator. Rather, it operates directly on any register in the register file, which results in significant improvement in execution time compared to accumulator-oriented devices.

The 8096BH supports a variety of options to simplify memory systems, interfacing requirements and ready control. Bus flexibility is provided by allowing selection of bus control signal definitions and run-time selection of the external bus width. In addition, several ready control modes are available to simplify the external hardware requirements for accessing slow devices.

Timers and HSIO Unit

Two 16-bit timers are available for use on the 8096BH. Timer 1 is used to synchronize events to real time, while timer 2 can be clocked externally and synchronizes events to external occurrences. Both timers can be used to trigger interrupts and set flags in the I/O status register.

The high-speed I/O (HSIO) unit works in conjunction with the timers. The input portion of the HSIO unit records the value on timer 1 when external transitions occur on any of four input pins, while the output section causes events (e.g., output pin transition, start A/D conversion, etc.) to occur based on values of either timer 1 or timer 2. The time resolution of the HSIO unit is 2μs.

An additional 16-bit timer, the watchdog timer, can be started under software control and counts state times. If not cleared before it overflows, the $\overline{RESET}$ pin will be asserted, causing the system to be reinitialized. This mechanism is provided as a means of recovery from a software upset.

I/O Ports

Five 8-bit ports are available in the 8096BH, in addition to the HSIO inputs and outputs.

Port 0 is an input-only port which also serves as the analog input port for those chip variants which include the A/D converter. Port 1 is a quasi-bidirectional I/O port which can be used as an input port, an output port, or mixed. Port 2 is like port 1, but can also be programmed to serve special functions such as serial input and output. Ports 3 and 4 are bidirectional I/O ports with open drain outputs. These ports are also used as the multiplexed address/data bus when accessing external memory. In this mode they can be configured to be either a multiplexed 16-bit address/data bus or a multiplexed 16-bit address/8-bit data bus.

The serial port is compatible with the serial port of 8051-type microcontrollers such as the 8031, the 80C51 and the 80C451. If features three asynchronous modes of operation and one synchronous mode. Bit rates in all modes are provided by an on-chip programmable bit rate generator.

A/D Converter and PW Modulator

The A/D converter in the 8X95BH and 8X97BH devices is a 10-bit, successive approximation converter with an internal sample and hold. When operating at 12MHz, a conversion requires 22μs, including 1μs for sampling. A conversion can be initiated either by setting a control bit in a command register or by programming the HSIO unit to trigger the conversion at some specified time.

The pulse-width modulator outputs a pulse train having a fixed period of 256 state times and a width programmable from 0 to 255 state times. This can be used to perform a digital to analog conversion.

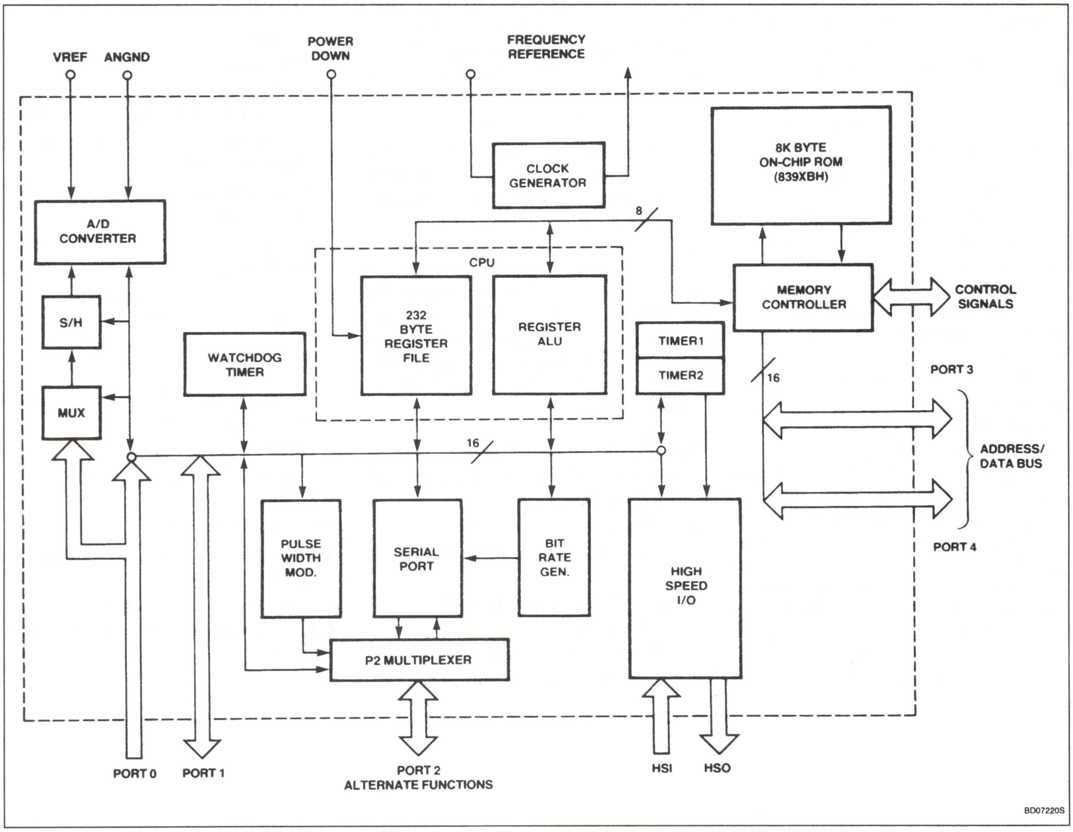
INSTRUCTION SET

The SC8096BH Series microcontrollers instruction repertoire includes 100 instructions, operating on a variety of data types which are likely to be useful in control-type applications. These include bits, bytes, words and double-words. Many of the instructions can be used with any one of six addressing modes. Instruction types include arithmetic, logical, data transfer and bit and stack operations. A full set of conditional jumps has been included to speed up testing for various conditions.

SC96BH Family

FEATURES	PACKAGES	ROMLESS	WITH ROM
Digital I/O only	48-pin	8094BH	8394BH
Digital I/O and 4-channel A/D	48-pin	8095BH	8395BH
Digital I/O only	68-pin	80967BH	83967BH
Digital I/O and 8-channel A/D	68-Pin	8097BH	8397BH

BLOCK DIAGRAM


VREF
ANGND
POWER DOWN
FREQUENCY REFERENCE
CLOCK GENERATOR
8K BYTE ON-CHIP ROM (839XBH)
8
A/D CONVERTER
CPU
232 BYTE REGISTER FILE
REGISTER ALU
MEMORY CONTROLLER
CONTROL SIGNALS
S/H
WATCHDOG TIMER
TIMER1
TIMER2
16
PORT 3
MUX
16
ADDRESS/ DATA BUS
PULSE WIDTH MOD.
SERIAL PORT
BIT RATE GEN.
HIGH SPEED I/O
PORT 4
P2 MULTIPLEXER
PORT 0
PORT 1
PORT 2 ALTERNATE FUNCTIONS
HSI
HSO
BD07220S

PCB5010/11
Single-Chip Digital Signal Processors

Preliminary Specification (Brief)

DESCRIPTION

The Signetics PCB5010 and PCB5011 are 16-bit single-chip programmable microcomputers with architectures optimized to perform the high-speed operations required for digital signal processing (DSP) and other numeric processing applications. Featuring a highly parallel architecture and extensive I/O capabilities, the PCB5010 and PCB5011 support applications in areas such as telecommunications, modems, image processing, speech and audio processing, digital filtering, spectrum analysis, and high-speed control.

The structure of these processors is characterized by a dual data bus, a two-operand hardware multiply-accumulate unit, and a two-operand arithmetic-logic unit (ALU) to improve throughput. Powerful parallel and serial interfaces enable communication with external devices. Internal data storage is provided via two 128-word data RAMs, each with an address computation unit that allows a wide variety of memory addressing schemes to be implemented. The PCB5010 also incorporates a 512-word coefficient or constant ROM, with its own address computation unit, and mask programmable ROM for program storage, making this device ideal for systems requiring low chip count. The PCB5011, a ROMless version, is available for small scale production runs and for situations where a large program memory (up to 64K) is required. The PCB5010 is packaged in a 68-pin plastic leaded chip carrier (PLCC) and the PCB5011 in a 144-lead pin grid array (PGA).

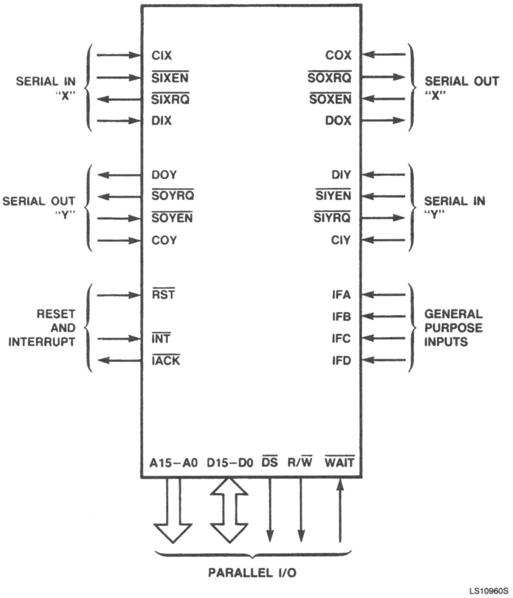
Instructions which control the operation of the PCB5010 and PCB5011 are stored in on-chip program memory and external memory, respectively. Each instruction comprises 40 bits arranged in several fields that specify the basic operations to be performed during instruction execution. Fabricated in CMOS technology, the PCB5010/11 utilize a limited amount of pipelining to allow the execution of a new instruction to begin every 125ns when operating with an 8MHz clock. In concert with the highly parallel architecture of these DSPs, this allows for the execution of up to 48 million basic operations per second.

Programming of the PCB5010/11 is simplified by use of a Pascal-like assembly language syntax and powerful development tools. These include a macro assembler with a library of standard routines, and a software simulator. These are available for installation in a number of different host computers. A stand-alone debug system (SDS) features real-time emulation capability and assists in the integration of the software with the target hardware.

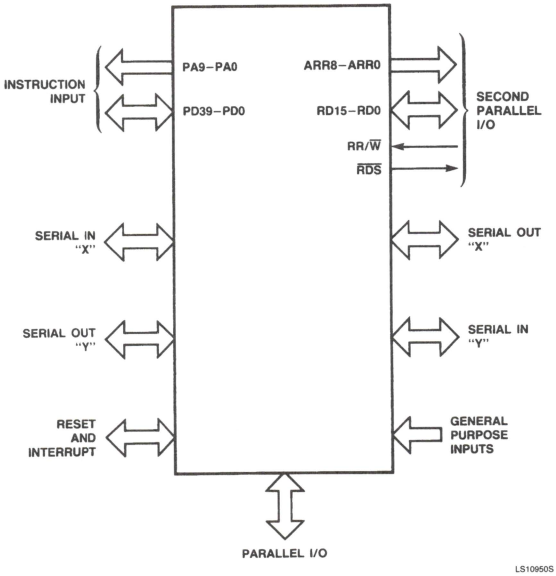
FEATURES

- **Harvard architecture with two 16-bit data busses**
- **Highly parallel operation**
  - **Up to six simultaneous sub-functions**
- **16 × 16 multiplier with 40-bit accumulator**
  - **45 different operations**
  - **Full range barrel shifter**
  - **Hardware support for multiple-precision operations**
- **16-bit, 2-operand ALU**
  - **31 operations**
  - **Hardware support for multiple-precision operations**
- **Four basic instruction types**
  - **Multiply-accumulate**
  - **ALU operation**
  - **Load immediate**
  - **Branch**
- **Program memory:**
  - **Internal 992 × 40 ROM + 32 × 40 RAM (PCB5010)**
  - **External 1K × 40 expandable to 64K × 40 (PCB5011)**
- **Data memory:**
  - **2 × (128 × 16) RAM + 512 × 16 ROM (ROM external on PCB5011)**
  - **Three programmable address computation units**
  - **Up to 64K × 16 external memory**
- **Software extendable 5-level stack**
- **16-bit I/O - up to 8 million words/sec**
- **Two full-duplex serial I/O channels**
- **Maskable interrupt**
- **8MHz clock, 125ns instruction cycle**
- **Low power CMOS technology**
- **Full complement of development support tools available**

EXTERNAL INTERFACES

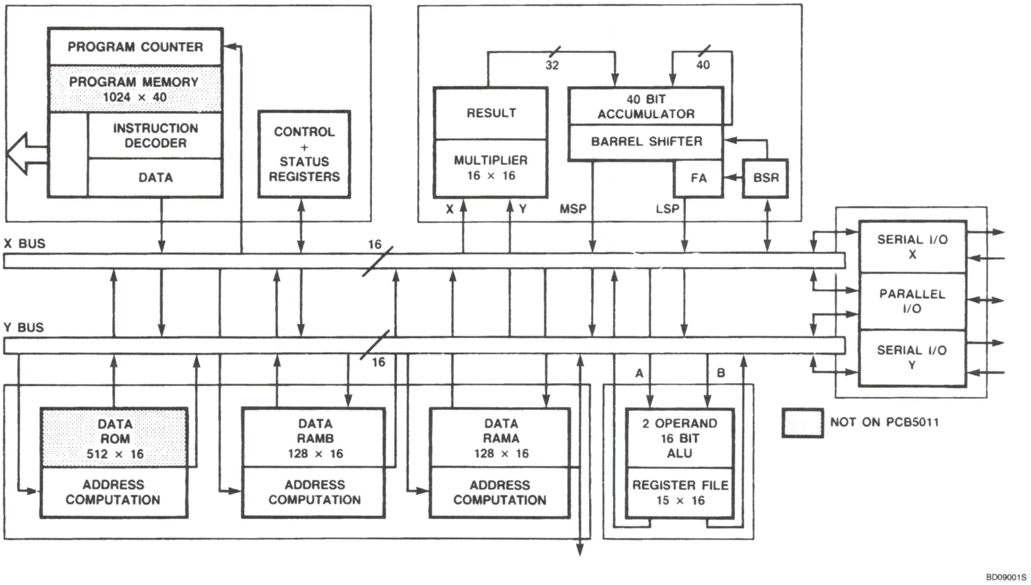

SERIAL IN "X"
CIX
SIXEN
SIXRQ
DIX
COX
SOXRQ
SOXEN
DOX
SERIAL OUT "X"
SERIAL OUT "Y"
DOY
SOYRQ
SOYEN
COY
DIY
SIYEN
SIYRQ
CIY
SERIAL IN "Y"
RESET AND INTERRUPT
RST
INT
IACK
IFA
IFB
IFC
IFD
GENERAL PURPOSE INPUTS
A15–A0
D15–D0
DS
R/W
WAIT
PARALLEL I/O
LS10960S

PCB5010 (68-pin PLCC)


INSTRUCTION INPUT
PA9–PA0
PD39–PD0
ARR8–ARR0
RD15–RD0
RR/W
RDS
SECOND PARALLEL I/O
SERIAL IN "X"
SERIAL OUT "X"
SERIAL OUT "Y"
SERIAL IN "Y"
RESET AND INTERRUPT
GENERAL PURPOSE INPUTS
PARALLEL I/O
LS10950S

PCB5011 (144-lead PGA)

Single-Chip Digital Signal Processors PCB5010/11


PROGRAM COUNTER
PROGRAM MEMORY 1024 × 40
INSTRUCTION DECODER
DATA
CONTROL + STATUS REGISTERS
RESULT
MULTIPLIER 16 × 16
32
40
40 BIT ACCUMULATOR
BARREL SHIFTER
FA
BSR
X
Y
MSP
LSP
X BUS
16
Y BUS
16
SERIAL I/O X
PARALLEL I/O
SERIAL I/O Y
DATA ROM 512 × 16
ADDRESS COMPUTATION
DATA RAMB 128 × 16
ADDRESS COMPUTATION
DATA RAMA 128 × 16
ADDRESS COMPUTATION
A
B
2 OPERAND 16 BIT ALU
REGISTER FILE 15 × 16
NOT ON PCB5011
BD09001S

PCB5010/11 Simplified Block Diagram

Single-Chip Digital Signal Processors

PCB5010/11

OPERATION

In the PCB5010/11, five major functional units — a multiplier-accumulator, an ALU and register file, data memories, serial and parallel I/O and program control unit — are interconnected via two 16-bit data busses, creating multiple parallel transfer paths. Additional dedicated, local busses within the functional units (e.g., between the multiplier and the accumulator and between the ALU and register file) eliminate congestion on the data busses and thus allows high throughput to be maintained. A 40-bit wide instruction word, stored internally in the PCB5010 and externally in the PCB5011, is subdivided into function-control fields which permits the corresponding functional units to be controlled in parallel. The devices are produced using a high-speed CMOS process, and their pipelined architecture permits a new instruction execution to start every 125ns.

These implementation features yield a cost effective device containing a well orchestrated set of resources working harmoniously to execute the sophisticated signal processing algorithms used in telecommunications, speech and image processing, and general industry applications. Either version can replace multiple ICs in current designs, affording a single-chip solution that makes many applications practical for the first time.

Program Control Unit

The program control unit in the PCB5010 contains the program memory, the program counter (PC), the instruction register (IR) and instruction decoder, the instruction repeat register, a hardware stack, a number of status and control registers and the maskable interrupt and reset logic. The control unit of the PCB5011 is identical, except that instead of the internal program memory it contains the interface to the external program memory.

The PCB5010 includes 987 words of mask-programmable ROM and 32 words of RAM to store the 40-bit instruction words. The program RAM, whose contents can be changed dynamically, can be loaded from external memory via the parallel input, the X-bus and a logic block which concatenates several of the 16-bit external words into the required 40 bits. A simple five-word RAM loading program is included in the program memory. The PCB5011 can directly access 1K instructions in external memory and operates at full speed. The addressing range can be extended to 64K by adding a small amount of external logic to the DSP.

There are two registers containing status and control information that can be accessed by the programmer. The program status register indicates the status of the accumulator, barrel-shifter, ALU, ACUs, parallel input port, and interrupt logic. It also defines whether instruction execution is in pipelined or non-pipelined mode. The serial I/O status and flag register contains information on the status of the serial I/O circuitry and reflects the signal levels at four user input pins, polling them every clock cycle.

The DSP supports a single maskable interrupt input. Occurrence of an interrupt automatically stores the return address on the stack and vectors the program to address 1. An interrupt acknowledge output signal informs the source when the interrupt is accepted. The 5-level hardware stack supports the nesting of interrupts and subroutines. The top of the stack is defined as a source and destination for the X-bus, so that the number of stack levels can be increased under software control. Additionally, the program status and other pertinent registers are also accessible via the data busses, so that their contents can also be stored on the stack or in data memory if required.

Program flow can be altered via software. Flags in status registers, or logical combinations of them, can be used as branching conditions for conditional jumps.

The instruction repeat register (RPR) allows the next sequential instruction to be repeated a specified number of times, from 3 to 255, without any additional program code.

Multiplier-Accumulator Unit (MAC)

The MAC provides a hardware implementation of various subfunctions which are used extensively when executing certain classes of vector product operations. The major elements in this unit are a 16 × 16-bit multiplier, using a modified Booth algorithm and carry-look-ahead techniques, and a 40-bit accumulator.

The multiplier delivers a full 32-bit product that is stored in the product register (PR). The accumulator inputs are PR and the output of the sign and scale-down unit (S/SD), which provides hardware support for negative and positive accumulations in either single or multiple precision. The 40-bit result of the accumulation is stored in the accumulator register (ACR). Since there are eight extra bits in the accumulator, at least 256 product accumulations can be performed without loss of accuracy. A status bit is set if accumulator overflow occurs.

The MAC includes circuitry to facilitate the post-scaling and selection of product accumulation results without the use of data-dependent single or multi-bit shifts. This consists of the barrel shifter, the format adjuster, the barrel-shift control register (BSR) and an out-of-range detection unit, all located at the output of the ACR. The barrel shifter extracts 32 contiguous bits from the 40-bit contents of the ACR. The programmer determines which group of 32 bits (E31-E0) is to be extracted by placing a value in the BSR. The out-of-range detection unit sets a status bit when the sign bit of the extracted word has no significance, viz., when one or more bits of ACR to the left of the extracted word differs from E31.

E31 through E0 are split into two groups. The most significant part (E31-E16) is designated MSP and can be routed directly to the X and Y data busses. The least significant part (E15-E0) passes through the format adjuster, which can alter the format of this word in several ways. The output is designated LSP and can also be routed to the X and Y data busses.

Selecting the two operands via the input logic, performing the multiplication, and storing the result in PR takes one basic processor cycle. In the next cycle that value is accumulated, the result stored in the ACR, 32 bits of the result are extracted and the format adjustment is performed. Hence, there is only one pipeline delay from input to output of the MAC. This means that the unit can produce a new result at the end of each clock cycle but the result is delayed for one clock cycle due to the initial loading of the pipeline.

Arithmetic Logic Unit and Register File

Comprising a 2-operand 16-bit ALU and a scratchpad file of fifteen 16-bit registers, this unit can perform 31 different operations. These include, in addition to the operations found in typical microcontrollers, a number of extended operations designed for multi-precision calculations. Each ALU operation may set (or clear) one or more status flags which can be tested and used for branching. The result of the operation will be placed in a designated location in the register file, or if only the status bits of the ALU are of interest, it will be discarded.

The three-port register file has the capability of reading two operands and writing one operand in the same cycle. Thus, the same register can be selected as the source and destination of an operation.

The architecture of this functional unit allows operations to be performed considerably faster and with fewer instructions than in other architectures which utilize a single operand ALU. The PCB5010/11 can, for example, add the contents of data memory A to the contents of data memory B and place the result in one of the scratchpad registers in a single cycle. This typically requires at least three instructions in other DSPs: 1) move data memory A to accumulator, 2) add data memory B to accumulator, 3) move accumulator to scratchpad.

Data Memories and Address Computation Units

The PCB5010 contains two 128 × 16 data RAMs and a 512 × 16 data ROM, each with its own address computation unit (ACU). Although the PCB5011 contains no data ROM, it has the RAMs and their ACUs and retains the ROM's ACU so that an external data memory, accessed via a dedicated 16-bit parallel port, can be used. This port is bidirectional and thus can also be used for write operations. In either version, an additional 64K words of data memory can be accessed via the parallel I/O port. The output of each memory is sampled each clock cycle and stored in an associated data register from where it can be sourced onto the X and/or Y busses.

Each ACU performs identically, although they have different word lengths to accommodate the different memory sizes. The ACU contains four registers which are loaded from the busses and a control unit that performs several arithmetic operations. Certain bits in the address register can be masked so that they will not be changed during ACU operation, easing modulo addressing and memory paging. The ACU also allows increment, decrement, and step operations to be performed. Furthermore, a 'bit-reverse' output from the ACU can be selected to support Fast Fourier Transform (FFT) algorithms.

Serial and Parallel I/O

The PCB5010 and PCB5011 have a 16-bit parallel and four independent serial interfaces.

Sixteen address pins are provided to allow the parallel interface to access 64K 16-bit words of external data memory, or peripheral devices mapped onto this address space. The addressing capability is divided into a 12-bit address produced by ACUA and a 4-bit page word generated by a page register which can be written to and read via the Y-bus. Parallel I/O transfers at full speed (8 million word/sec) are possible provided the memory access time is short enough (about 60ns). When transmission speed is not critical, slower and therefore less expensive external memory can be used since a WAIT pin has been provided to allow the external memory to delay the processor, if necessary, while it stores or sends information. For the programmer, it makes no difference whether the DSPs are using internal memory or external memory with or without WAIT states.

The serial interfaces consist of two inputs and two outputs which can operate from DC to a maximum of 4 million bits/sec. Transfer of serial data is completely controlled by external devices that, in addition to other signals, provide the shift clocks. These transfers occur asynchronously without any on-chip processing other than loading and reading the data. Status bits are set to inform the program control unit when data is available to be read and when new data can be loaded for serialization. The word length of data received or transmitted via the serial interfaces can be programmed at any length from one-to-sixteen bits. The serial I/O ports can be used to implement a multi-processing system or can be used to provide data transfers to and from serial devices, as in a telecommunications system.

INSTRUCTIONS

The 40-bit instructions are designed to permit simultaneous control of the functional units and to avoid sequential-control delays. There are four basic instruction types, each leading to the execution of one or more (up to six) basic operations. Although the instructions support simultaneous functional operation, there are no complex instructions for the programmer to remember, since each instruction is divided into fields containing information for controlling the functional units. Thus, the Type 0 instruction is comprised of an ALU operation, up to two MOVE operations on the data busses and up to three ACU (address computation and data memory read) operations. The Type 1 instruction is identical, except that a MAC (multiply/accumulate) operation takes the place of the ALU operation. A Type 2 instruction combines a BRANCH operation with up to three ACU operations, while a Type 3 instruction encompasses a LOAD IMMEDIATE operation and up to three ACU operations. The programming language and assembler relieve the programmer from the tedious task of specifying each field separately, and automatically assign use of the X and Y-busses as appropriate.

The basic operations are similar to but typically more powerful than normal micro-processor instructions. They offer many options for programming, including 45 different multiply/accumulate operations, 31 types of ALU operations, eight types of ACU address computations, and four types of branches with 50 branching conditions.

DEVELOPMENT SUPPORT

A complete set of tools are available to aid program development and firmware/hardware integration.

Software tools, available for both VAX/VMS[1] and MS-DOS[1] systems, include an assembler, a simulator, and a macro library. The assembly language supports writing of programs in a Pascal-like structure and simplifies the task of programming the wide instructions. It supports both user-defined and library macros and produces output files for the simulator, for the hardware support tools, for PROM programming, and for mask generation for the program and data ROMs. The simulator accurately emulates the operation of the DSP. It allows all internal registers to be examined and modified, and can operate in both single step and run modes. Several breakpoints are available, each of which can be triggered upon occurrence of a user-defined set of conditions. Both serial and parallel I/O functions can be simulated. Unique features of the simulator include an extensive on-line help facility and the ability to define the screen display of the device's resources in a manner most meaningful to the user. Users may also define their own simulation instructions. It is also possible to extend the scope of the simulator so as to simulate its environment, even to the extent of simulating multiple DSP systems. The macro library assists the programmer by providing pre-tested routines for a variety of arithmetic operations, filters, and FFT functions.

The hardware devices include a prototyping board and a stand alone development system (SDS). The former consists of a PC board containing a PCB5011 and high speed RAM to replace the program and data ROMs of a PCB5010. The RAMs may be loaded via a serial RS-232 link or from on-board PROMs. The SDS provides real-time, fully transparent emulation and program simulation. It links to a host processor via a serial RS-232 link for uploading and downloading of code and includes a variety of instructions to assist debugging of programs and firmware/hardware integration. Real-time hardware breakpoints are provided, as well as a trace memory to monitor the execution of up to 2K program steps.

---

[1] VAX/VMS is a trademark of Digital Equipment Corporation. MS-DOS is a registered trademark of Microsoft Corporation.

Signetics

I^{2}C Bus Specification

Microcontroller Products

INTRODUCTION

For 8-bit applications, such as those requiring single-chip microcomputers, certain design criteria can be established:

- **A complete system usually consists of at least one microcomputer and other peripheral devices, such as memories and I/O expanders.**
- **The cost of connecting the various devices within the system must be kept to a minimum.**
- **Such a system usually performs a control function and does not require high-speed data transfer.**
- **Overall efficiency depends on the devices chosen and the interconnecting bus structure.**

In order to produce a system to satisfy these criteria, a serial bus structure is needed. Although serial buses don't have the throughput capability of parallel buses, they do require less wiring and fewer connecting pins. However, a bus is not merely an interconnecting wire, it embodies all the formats and procedures for communication within the system.

Devices communicating with each other on a serial bus must have some form of protocol which avoids all possibilities of confusion, data loss and blockage of information. Fast devices must be able to communicate with slow devices. The system must not be dependent on the devices connected to it, otherwise modifications or improvements would be impossible. A procedure has also to be resolved to decide which device will be in control of the bus and when. And if different devices with different clock speeds are connected to the bus, the bus clock source must be defined.

All these criteria are involved in the specification of the I^{2}C bus.

THE I^{2}C BUS CONCEPT

Any manufacturing process (NMOS, CMOS, I^{2}L) can be supported by the I^{2}C bus. Two wires (SDA – serial data, SCL – serial clock) carry information between the devices connected to the bus. Each device is recognized by a unique address – whether it is a microcomputer, LCD driver, memory or keyboard interface – and can operate as either a transmitter or receiver, depending on the function of the device. Obviously an LCD driver is only a receiver, while a memory can both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers (see Table 1). A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

The I^{2}C bus is a multi-master bus. This means that more than one device capable of controlling the bus can be connected to it. As masters are usually microcomputers, let's consider the case of a data transfer between two microcomputers connected to the I^{2}C bus (Figure 1). This highlights the master-slave and receiver-transmitter relationships to be found on the I^{2}C bus. It should be noted that these relationships are not permanent, but only depend on the direction of data transfer at that time. The transfer of data would follow in this way:

1) Suppose microcomputer A wants to send information to microcomputer B
   - microcomputer A (master) addresses microcomputer B (slave)
   - microcomputer A (master transmitter) sends data to microcomputer B (slave receiver)
   - microcomputer A terminates the transfer.
2) If microcomputer A wants to receive information from microcomputer B
   - microcomputer A (master) addresses microcomputer B (slave)
   - microcomputer A (master receiver) receives data from microcomputer B (slave transmitter)
   - microcomputer A terminates the transfer.

Even in this case, the master (microcomputer A) generates the timing and terminates the transfer.

The possibility of more than one microcomputer being connected to the I^{2}C bus

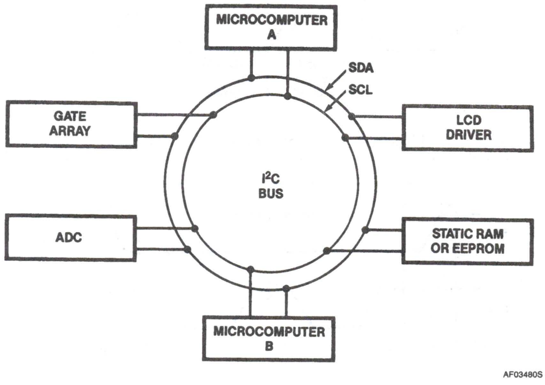


Figure 1. Typical I^{2}C Bus Configuration

I^{2}C Bus Specification

Table 1. Definition of I^{2}C Bus Terminology

TERM	DESCRIPTION
Transmitter	The device which sends data to the bus
Receiver	The device which receives data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by a master
Multi-master	More than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	Procedure to ensure that if more than one master simultaneously tries to control the bus, only one is allowed to do so and the message is not corrupted
Synchronization	Procedure to synchronize the clock signals of two or more devices

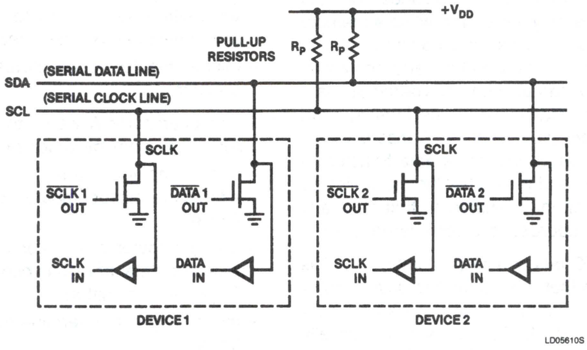


Figure 2. Connection of Devices to the I^{2}C Bus

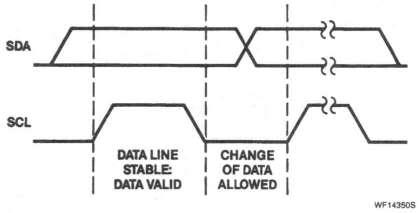


Figure 3. Bit Transfer on the I^{2}C Bus

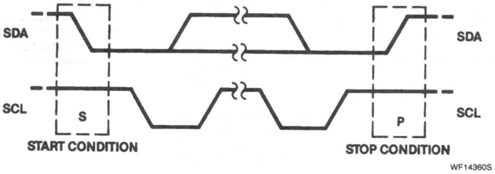


Figure 4. Start and Stop Conditions

means that more than one master could try to initiate a data transfer at the same time. To avoid the chaos that might ensue from such an event, an arbitration procedure has been developed. This procedure relies on the wired-AND connection of all devices to the I^{2}C bus.

If two or more masters try to put information on to the bus, the first to produce a one when the other produces a zero will lose the arbitration. The clock signals during arbitration are a synchronized combination of the clocks generated by the masters using the wired-AND connection to the SCL line (for more detailed information concerning arbitration see Arbitration and Clock Generation).

Generation of clock signals on the I^{2}C bus is always the responsibility of master devices; each master generates its own clock signals when transferring data on the bus. Bus clock signals from a master can only be altered when they are stretched by a slow slave device holding down the clock line or by another master when arbitration takes place.

GENERAL CHARACTERISTICS

Both SDA and SCL are bidirectional lines, connected to a positive supply voltage via a pull-up resistor (see Figure 2). When the bus is free, both lines are High. The output stages of devices connected to the bus must have an open-drain or open-collector in order to perform the wired-AND function. Data on the I^{2}C bus can be transferred at a rate up to 100kbit/s. The number of devices connected to the bus is solely dependent on the limiting bus capacitance of 400pF.

BIT TRANSFER

Due to the variety of different technology devices (CMOS, NMOS, I^{2}L) which can be connected to the I^{2}C bus, the levels of the logical 0 (Low) and 1 (High) are not fixed and depend on the appropriate level of V$_{DD}$ (see Electrical Specifications). One clock pulse is generated for each data bit transferred.

Data Validity

The data on the SDA line must be stable during the High period of the clock. The High or Low state of the data line can only change when the clock signal on the SCL line is Low (Figure 3).

Start and Stop Conditions

Within the procedure of the I^{2}C bus, unique situations arise which are defined as start and stop conditions (see Figure 4).

A High-to-Low transition of the SDA line while SCL is High is one such unique case. This situation indicates a start condition.

I^{2}C Bus Specification

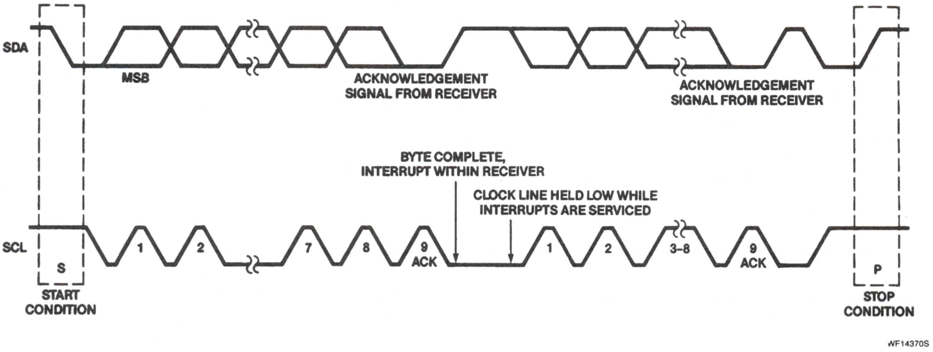

SDA
MSB
ACKNOWLEDGEMENT SIGNAL FROM RECEIVER
ACKNOWLEDGEMENT SIGNAL FROM RECEIVER
BYTE COMPLETE, INTERRUPT WITHIN RECEIVER
CLOCK LINE HELD LOW WHILE INTERRUPTS ARE SERVICED
SCL
S
1
2
7
8
9
ACK
1
2
3-8
9
ACK
P
START CONDITION
STOP CONDITION
WF14370S

Figure 5. Data Transfer on the I^{2}C Bus

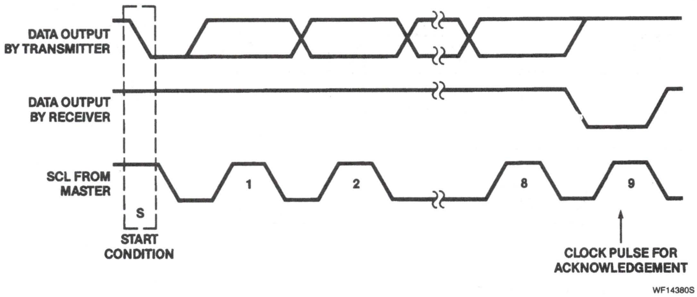

DATA OUTPUT BY TRANSMITTER
DATA OUTPUT BY RECEIVER
SCL FROM MASTER
S
1
2
8
9
START CONDITION
CLOCK PULSE FOR ACKNOWLEDGEMENT
WF14380S

Figure 6. Acknowledge on the I^{2}C Bus

A Low-to-High transition of the SDA line while SCL is High defines a stop condition.

Start and stop conditions are always generated by the master. The bus is considered to be busy after the start condition. The bus is considered to be free again a certain time after the stop condition. This bus free situation will be described later in detail.

Detection of start and stop conditions by devices connected to the bus is easy if they possess the necessary interfacing hardware. However, microcomputers with no such interface have to sample the SDA line at least twice per clock period in order to sense the transition.

TRANSFERRING DATA

Byte Format

Every byte put on the SDA line must be 8 bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte must be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first (Figure 5). If a receiving device cannot receive another complete byte of data until it has performed some other function, for example, to service an internal interrupt, it can hold the clock line SCL Low to force the transmitter into a wait state. Data transfer then continues when the receiver is ready for another byte of data and releases the clock line SCL.

In some cases, it is permitted to use a different format from the I^{2}C bus format, such as CBUS compatible devices. A message which starts with such an address can be terminated by the generation of a stop condition, even during the transmission of a byte. In this case, no acknowledge is generated.

Acknowledge

Data transfer with acknowledge is obligatory. The acknowledge-related clock pulse is generated by the master. The transmitting device releases the SDA line (High) during the acknowledge clock pulse.

The receiving device has to pull down the SDA line during the acknowledge clock pulse so that the SDA line is stable Low during the high period of this clock pulse (Figure 6). Of course, setup and hold times must also be taken into account and these will be described in the Timing section.

Usually, a receiver which has been addressed is obliged to generate an acknowledge after each byte has been received (except when the message starts with a CBUS address.

When a slave receiver does not acknowledge on the slave address, for example, because it is unable to receive while it is performing some real-time function, the data line must be left High by the slave. The master can then generate a STOP condition to abort the transfer.

If a slave receiver does acknowledge the slave address, but some time later in the transfer cannot receive any more data bytes, the master must again abort the transfer. This is indicated by the slave not generating the acknowledge on the first byte following. The slave leaves the data line High and the master generates the STOP condition.

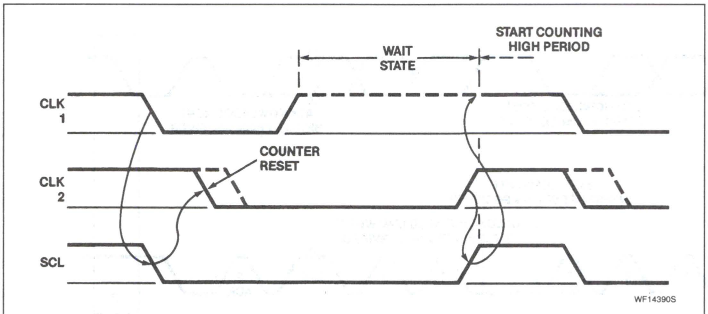


Figure 7. Clock Synchronization During the Arbitration Procedure

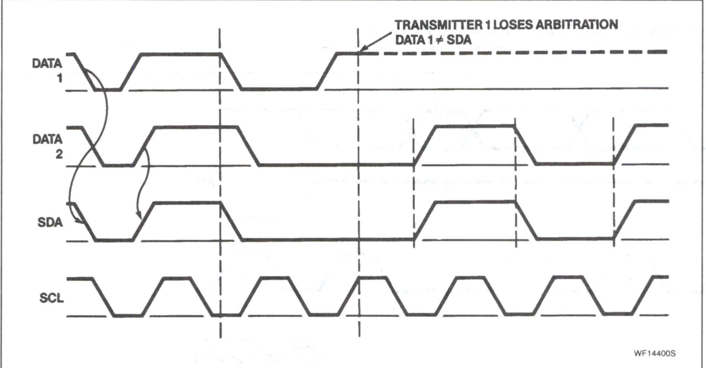


Figure 8. Arbitration Procedure of Two Masters

In the case of a master receiver involved in a transfer, it must signal an end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate the STOP condition.

ARBITRATION AND CLOCK GENERATION

Synchronization

All masters generate their own clock on the SCL line to transfer messages on the I^{2}C bus. Data is only valid during the clock High period on the SCL line; therefore, a defined clock is needed if the bit-by-bit arbitration procedure is to take place.

Clock synchronization is performed using the wired-AND connection of devices to the SCL LINE. This means that a High-to-Low transition on the SCL line will affect the devices concerned, causing them to start counting off their Low period. Once a device clock has gone Low it will hold the SCL line in that state until the clock High state is reached (Figure 7). However, the Low-to-High change in this device clock may not change the state of the SCL line if another device clock is still within its Low period. Therefore, SCL will be held Low by the device with the longest Low period. Devices with shorter Low periods enter a High wait state during this time.

When all devices concerned have counted off their Low period, the clock line will be released and go High. There will then be no difference between the device clocks and the state of the SCL line and all of them will start counting their High periods. The first device to complete its High period will again pull the SCL line Low.

In this way, a synchronized SCL clock is generated for which the Low period is determined by the device with the longest clock Low period while the High period on SCL is determined by the device with the shortest clock High period.

Arbitration

Arbitration takes place on the SDA line in such a way that the master which transmits a High level, while another master transmits a Low level, will switch off its DATA output stage since the level on the bus does not correspond to its own level.

Arbitration can carry on through many bits. The first stage of arbitration is the comparison of the address bits. If the masters are each trying to address the same device, arbitration continues into a comparison of the data. Because address and data information is used on the I^{2}C bus for the arbitration, no information is lost during this process.

A master which loses the arbitration can generate clock pulses until the end of the byte in which it loses the arbitration.

If a master does lose arbitration during the addressing stage, it is possible that the winning master is trying to address it. Therefore, the losing master must switch over immediately to its slave receiver mode.

Figure 8 shows the arbitration procedure for two masters. Of course more may be involved, depending on how many masters are connected to the bus. The moment there is a difference between the internal data level of the master generating DATA 1 and the actual level on the SDA line, its data output is switched off, which means that a High output level is then connected to the bus. This will not affect the data transfer initiated by the winning master. As control of the I^{2}C bus is decided solely on the address and data sent by competing masters, there is no central master, nor any order of priority on the bus.

Use of the Clock Synchronizing Mechanism as a Handshake

In addition to being used during the arbitration procedure, the clock synchronization mechanism can be used to enable receiving devices to cope with fast data transfers, either on a byte or bit level.

On the byte level, a device may be able to receive bytes of data at a fast rate, but needs more time to store a received byte or prepare another byte to be transmitted. Slave devices can then hold the SCL line Low, after reception and acknowledge of a byte, to force the master into a wait state until the slave is ready for the next byte transfer in a type of handshake procedure.

On the bit level, a device such as a microcomputer without a hardware I^{2}C interface on-chip can slow down the bus clock by extending each clock Low period. In this way, the speed of any master is adapted to the internal operating rate of this device.

I^{2}C Bus Specification

FORMATS

Data transfers follow the format shown in Figure 9. After the start condition, a slave address is sent. This address is 7 bits long; the eighth bit is a data direction bit (R/$\overline{W}$). A zero indicates a transmission (WRITE); a one indicates a request for data (READ). A data transfer is always terminated by a stop condition generated by the master. However, if a master still wishes to communicate on the bus, it can generate another start condition, and address another slave without first generating a stop condition. Various combinations of read/write formats are then possible within such a transfer.

At the moment of the first acknowledge, the master transmitter becomes a master receiver and the slave receiver becomes a slave transmitter. This acknowledge is still generated by the slave.

The stop condition is generated by the master.

During a change of direction within a transfer, the start condition and the slave address are both repeated, but with the R/$\overline{W}$ bit reversed.

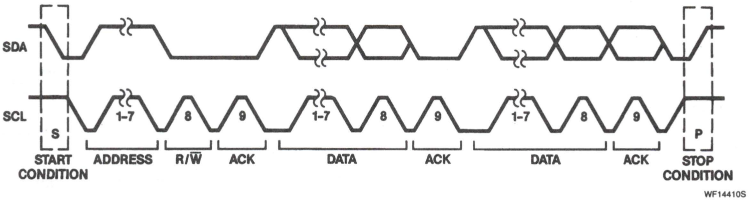
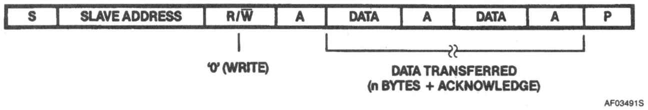

SDA
SCL
S
1-7
8
9
1-7
8
9
1-7
8
9
P
START CONDITION
ADDRESS
R/W̄
ACK
DATA
ACK
DATA
ACK
STOP CONDITION
WF14410S

Figure 9. A Complete Data Transfer

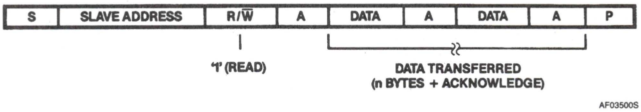
Possible Data Transfer Formats are:

a) Master transmitter transmits to slave receiver. Direction is not changed.

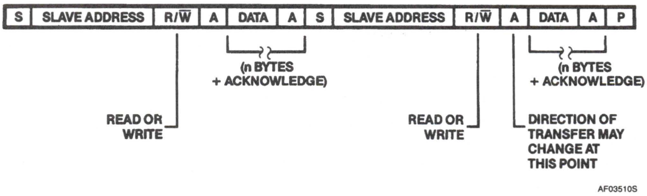
A = ACKNOWLEDGE
S = START
P = STOP


S
SLAVE ADDRESS
R/W̄
A
DATA
A
DATA
A
P
'0' (WRITE)
DATA TRANSFERRED
(n BYTES + ACKNOWLEDGE)
AF03491S

b) Master reads slave immediately after first byte.


S
SLAVE ADDRESS
R/W̄
A
DATA
A
DATA
A
P
'1' (READ)
DATA TRANSFERRED
(n BYTES + ACKNOWLEDGE)
AF03500S

c) Combined formats.


S
SLAVE ADDRESS
R/W̄
A
DATA
A
S
SLAVE ADDRESS
R/W̄
A
DATA
A
P
(n BYTES
+ ACKNOWLEDGE)
(n BYTES
+ ACKNOWLEDGE)
READ OR
WRITE
READ OR
WRITE
DIRECTION OF
TRANSFER MAY
CHANGE AT
THIS POINT
AF03510S

NOTES:

1. Combined formats can be used, for example, to control a serial memory. During the first data byte, the internal memory location has to be written. After the start condition is repeated, data can then be transferred.
2. All decisions on auto-increment or decrement of previously accessed memory locations, etc., are taken by the designer of the device.
3. Each byte is followed by an acknowledge as indicated by the A blocks in the sequence.
4. I^{2}C devices have to reset their bus logic on receipt of a start condition so that they all anticipate the sending of a slave address.

ADDRESSING

The first byte after the start condition determines which slave will be selected by the master. Usually, this first byte follows that start procedure. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge, although devices can be made to ignore this address. The second byte of the general call address then defines the action to be taken.

Definition of Bits in the First Byte

The first seven bits of this byte make up the slave address (Figure 10). The eighth bit (LSB – least significant bit) determines the direction of the message. A zero on the least significant position of the first byte means that the master will write information to a selected slave; a one in this position means that the master will read information from the slave.

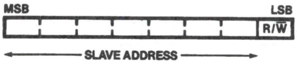


Figure 10. The First Byte After the Start Procedure

When an address is sent, each device in a system compares the first 7 bits after the start condition with its own address. If there is a match, the device will consider itself addressed by the master as a slave receiver or slave transmitter, depending on the $R/\overline{W}$ bit.

The slave address can be made up of a fixed and a programmable part. Since it is expected that identical ICs will be used more than once in a system, the programmable part of the slave address enables the maximum possible number of such devices to be connected to the I^{2}C bus. The number of programmable address bits of a device depends on the number of pins available. For example, if a device has 4 fixed and 3 programmable address bits, a total of eight identical devices can be connected to the same bus.

The I^{2}C bus committee is available to coordinate allocation of I^{2}C addresses.

The bit combination 1111XXX of the slave address is reserved for future extension purposes.

The address 1111111 is reserved as the extension address. This means that the addressing procedure will be continued in the next byte(s). Devices that do not use the extended addressing do not react at the reception of this byte. The seven other possibilities in group 1111 will also only be used for extension purposes but are not yet allocated.

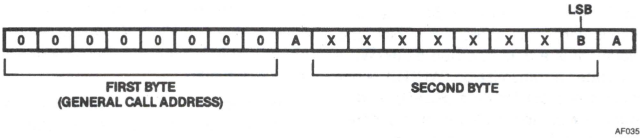


Figure 11. General Call Address Format

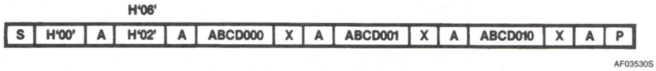


Figure 12. Sequence of a Programming Master

The combination 0000XXX has been defined as a special group. The following addresses have been allocated:

FIRST BYTE			
Slave Address		$R/\overline{W}$	
0000	000	0	General call address
0000	000	1	Start byte
0000	001	X	CBUS address
0000	010	X	Address reserved for different bus format
0000	011	X	To be defined
0000	100	X	
0000	101	X	
0000	110	X	
0000	111	X	

No device is allowed to acknowledge at the reception of the start byte.

The CBUS address has been reserved to enable the intermixing of CBUS and I^{2}C devices in one system. I^{2}C bus devices are not allowed to respond at the reception of this address.

The address reserved for a different bus format is included to enable the mixing of I^{2}C and other protocols. Only I^{2}C devices that are able to work with such formats and protocols are allowed to respond to this address.

General Call Address

The general call address should be used to address every device connected to the I^{2}C bus. However, if a device does not need any of the data supplied within the general call structure, it can ignore this address by not acknowledging. If a device does require data from a general call address, it will acknowledge this address and behave as a slave receiver. The second and following bytes will be acknowledged by every slave receiver capable of handling this data. A slave which cannot process one of these bytes must ignore it by not acknowledging.

The meaning of the general call address is always specified in the second byte (Figure 11).

There are two cases to consider:

1. When the least significant bit B is a zero.
2. When the least significant bit B is a one.

When B is a zero, the second byte has the following definition:

00000110 (H'06') Reset and write the programmable part of slave address by software and hardware. On receiving this two-byte sequence, all devices (designed to respond to the general call address) will reset and take in the programmable part of their address.

Precautions must be taken to ensure that a device is not pulling down the SDA or SCL line after applying the supply voltage, since these low levels would block the bus.

00000010 (H'02') Write slave address by software only. All devices which obtain the programmable part of their address by software (and which have been designed to respond to the general call address) will enter a mode in which they can be programmed. The device will not reset.

I^{2}C Bus Specification

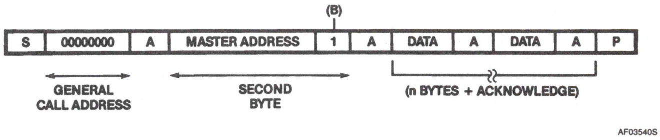
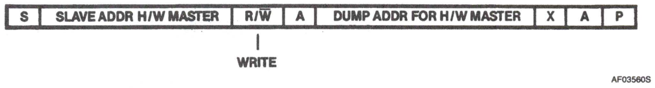
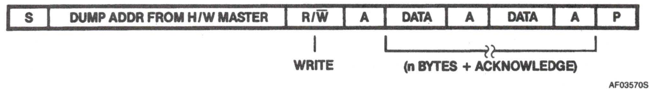


Figure 13. Data Transfer From Hardware Master Transmitter



a. Configuring master sends dump address to hardware master.



b. Hardware master dumps data to selected slave device.

Figure 14. Data Transfer of Hardware Master Transmitter Capable of Dumping Data Directly to Slave Devices

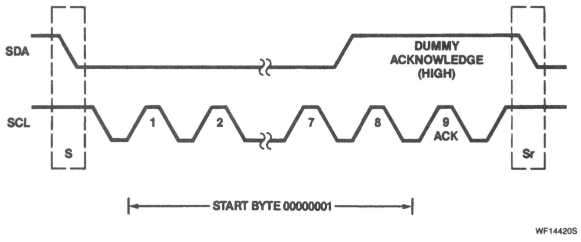


Figure 15. Start Byte Procedure

An example of a data transfer of a programming master is shown in Figure 12 (ABCD represents the fixed part of the address).

00000100 (H'04')	Write slave address by hardware only. All devices which define the programmable part of their address by hardware (and which respond to the general call address) will latch this programmable part at the reception of this two-byte sequence. The device will not reset.
00000000 (H'00')	This code is not allowed to be used as the second byte.

Sequences of programming procedure are published in the appropriate device data sheets.

The remaining codes have not been fixed and devices must ignore these codes.

When B is a one, the two-byte sequence is a hardware general call. This means that the sequence is transmitted by a hardware master device, such as a keyboard scanner, which cannot be programmed to transmit a desired slave address. Since a hardware master does not know in advance to which device the message must be transferred, it can only generate this hardware general call and its own address, thereby identifying itself to the system (Figure 13).

The seven bits remaining in the second byte contain the device address of the hardware master. This address is recognized by an intelligent device, such as a microcomputer, connected to the bus which will then direct the information coming from the hardware master. If the hardware master can also act as a slave, the slave address is identical to the master address.

In some systems an alternative could be that the hardware master transmitter is brought in the slave receiver mode after the system reset. In this way, a system configuring master can tell the hardware master transmitter (which is now in slave receiver mode) to which address data must be sent (Figure 14). After this programming procedure, the hardware master remains in the master transmitter mode.

Start Byte

Microcomputers can be connected to the I^{2}C bus in two ways. If an on-chip hardware I^{2}C bus interface is present, the microcomputer can be programmed to be interrupted only by requests from the bus. When the device possesses no such interface, it must constantly monitor the bus via software. Obviously, the more times the microcomputer monitors, or polls, the bus, the less time it can spend carrying out its intended function.

Therefore, there is a difference in speed between fast hardware devices and the relatively slow microcomputer which relies on software polling.

In this case, data transfer can be preceded by a start procedure which is much longer than normal (Figure 15). The start procedure consists of:

a) A start condition, (S)
b) A start byte 00000001
c) An acknowledge clock pulse
d) A repeated start condition, (Sr)

After the start condition (S) has been transmitted by a master requiring bus access, the start byte (00000001) is transmitted. Another microcomputer can therefore sample the SDA line on a low sampling rate until one of the seven zeros in the start byte is detected. After detection of this Low level on the SDA line, the microcomputer is then able to switch to a higher sampling rate in order to find the second start condition (Sr) which is then used for synchronization.

A hardware receiver will reset at the reception of the second start condition (Sr) and will therefore ignore the start byte.

After the start byte, an acknowledge-related clock pulse is generated. This is present only to conform with the byte handling format used on the bus. No device is allowed to acknowledge the start byte.

I^{2}C Bus Specification

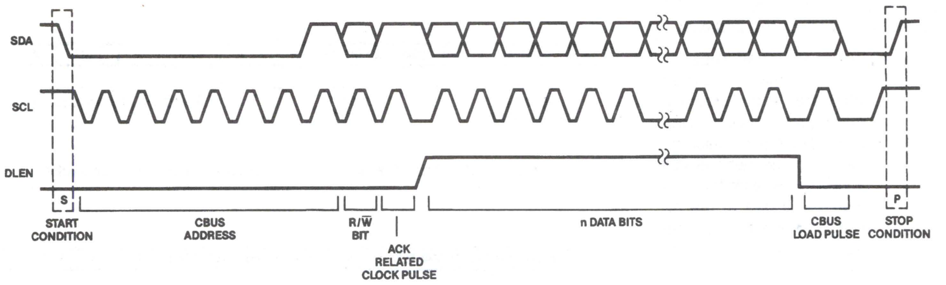


Figure 16. Data Format of Transmissions With CBUS Receiver/Transmitter

CBUS Compatibility
Existing CBUS receivers can be connected to the I^{2}C bus. In this case, a third line called DLEN has to be connected and the acknowledge bit omitted. Normally, I^{2}C transmissions are multiples of 8-bit bytes; however, CBUS devices have different formats.

In a mixed bus structure, I^{2}C devices are not allowed to respond on the CBUS message. For this reason, a special CBUS address (0000001X) has been reserved. No I^{2}C device will respond to this address. After the transmission of the CBUS address, the DLEN line can be made active and transmission, according to the CBUS format, can be performed (Figure 16).

After the stop condition, all devices are again ready to accept data.

Master transmitters are allowed to generate CBUS formats after having sent the CBUS address. Such a transmission is terminated by a stop condition, recognized by all devices. In the low speed mode, full 8-bit bytes must always be transmitted and the timing of the DLEN signal adapted.

If the CBUS configuration is known and no expansion with CBUS devices is foreseen, the user is allowed to adapt the hold time to the specific requirements of device(s) used.

ELECTRICAL SPECIFICATIONS OF INPUTS AND OUTPUTS OF I^{2}C DEVICES

The I^{2}C bus allows communication between devices made in different technologies which might also use different supply voltages.

For devices with fixed input levels, operating on a supply voltage of +5V ± 10%, the following levels have been defined:

V_{ILmax} = 1.5V (maximum input Low voltage)

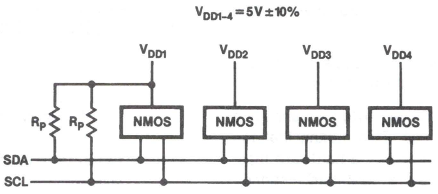


Figure 17. Fixed Input Level Devices Connected to the I^{2}C Bus

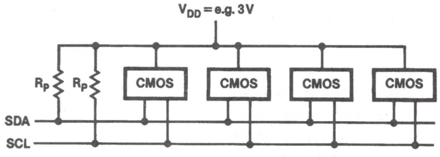


Figure 18. Devices With a Wide Range of Supply Voltages Connected to the I^{2}C Bus

V_{IHmin} = 3V (minimum input High voltage)

Devices operating on a fixed supply voltage different from +5V (e.g. I^{2}L), must also have these input levels of 1.5V and 3V for V_{IL} and V_{IH}, respectively.

For devices operating over a wide range of supply voltages (e.g. CMOS), the following levels have been defined:

V_{ILmax} = 0.3V_{DD} (maximum input Low voltage)
V_{IHmin} = 0.7V_{DD} (minimum input High voltage)

For both groups of devices, the maximum output Low value has been defined:

V_{OLmax} = 0.4V (max. output voltage Low) at 3mA sink current

The maximum low-level input current at V_{OLmax} of both the SDA pin and the SCL pin of an I^{2}C device is −10μA, including the leakage current of a possible output stage.

The maximum high-level input current at 0.9V_{DD} of both the SDA pin and SCL pin of an I^{2}C device is 10μA, including the leakage current of a possible output stage.

The maximum capacitance of both the SDA pin and the SCL pin of an I^{2}C device is 10pF.

Devices with fixed input levels can each have their own power supply of +5V ± 10%. Pull-up resistors can be connected to any supply (see Figure 17).

However, the devices with input levels related to V_{DD} must have one common supply line to which the pull-up resistor is also connected (see Figure 18).

I^{2}C Bus Specification

When devices with fixed input levels are mixed with devices with V_{DD}-related levels, the latter devices have to be connected to one common supply line of +5V ± 10% along with the pull-up resistors (Figure 19).

Input levels are defined in such a way that:

1. The noise margin on the Low level is 0.1 V_{DD}.
2. The noise margin on the High level is 0.2 V_{DD}.
3. Series resistors (R_S) up to 300Ω can be used for flash-over protection against high voltage spikes on the SDA and SCL line (due to flash-over of a TV picture tube, for example) (Figure 20).

The maximum bus capacitance per wire is 400pF. This includes the capacitance of the wire itself and the capacitance of the pins connected to it.

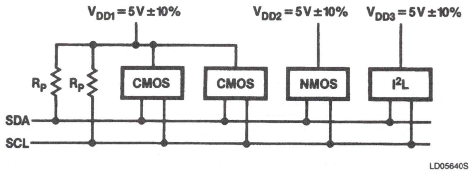


Figure 19. Devices With V_{DD} Related Levels Mixed With Fixed Input Level Devices on the I^{2}C Bus

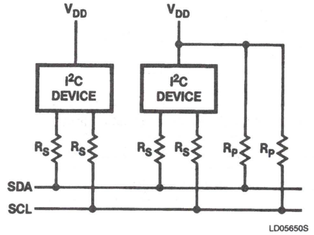


Figure 20. Serial Resistors (R_S) for Protection Against High Voltage

TIMING

The clock on the I^{2}C bus has a minimum Low period of 4.7μs and a minimum High period of 4μs. Masters in this mode can generate a bus clock with a frequency from 0 to 100kHz.

All devices connected to the bus must be able to follow transfers with frequencies up to 100kHz, either by being able to transmit or receive at that speed or by applying the clock synchronization procedure which will force the master into a wait state and stretch the Low periods. In the latter case the frequency is reduced.

Figure 21 shows the timing requirements in detail. A description of the abbreviations used is shown in Table 2. All timing references are at V_{ILmax} and V_{ILmin}.

LOW SPEED MODE

As explained previously, there is a difference in speed on the I^{2}C bus between fast hardware devices and the relatively slow microcomputer which relies on software polling. For this reason a low speed mode is available on the I^{2}C bus to allow these microcomputers to poll the bus less often.

Start and Stop Conditions

In the low speed mode, data transfer is preceded by the start procedure.

Data Format and Timing

The bus clock in this mode has a Low period of 130μs ± 25μs and a High period of 390μs ± 25μs, resulting in a clock frequency of approx. 2kHz. The duty cycle of the clock has this Low-to-High ratio to allow for more efficient use of microcomputers without an on-chip hardware I^{2}C bus interface. In this mode also, data transfer with acknowledge is obligatory. The maximum number of bytes transferred is not limited (Figure 22).

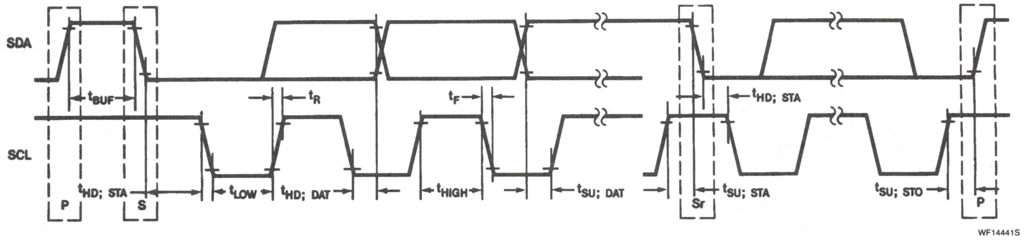


Figure 21. Timing Requirements for the I^{2}C Bus

Table 2. Timing Requirement for the I^{2}C Bus

SYMBOL	PARAMETER	LIMITS		UNIT
		Min	Max	
f_{SCL}	SCL clock frequency	0	100	kHz
t_{BUF}	Time the bus must be free before a new transmission can start	4.7		μs
$t_{HD;\ STA}$	Hold time start condition. After this period the first clock pulse is generated	4		μs
t_{Low}	The Low period of the clock	4.7		μs
t_{High}	The High period of the clock	4		μs
$t_{SU;\ STA}$	Setup time for start condition (Only relevant for a repeated start condition)	4.7		μs
$t_{HD;\ DAT}$	Hold time DATA for CBUS compatible masters for I^{2}C devices	 5 0*		 μs μs
$t_{SU;\ DAT}$	Setup time DATA	250		ns
t_R	Rise time of both SDA and SCL lines		1	μs
t_F	Fall time of both SDA and SCL lines		300	ns
$t_{SU;\ STO}$	Setup time for stop condition	4.7		μs

NOTES:

All values referenced to V_{IH} and V_{IL} levels.

* Note that a transmitter must internally provide a hold time to bridge the undefined region (300ns max.) of the falling edge of SCL.

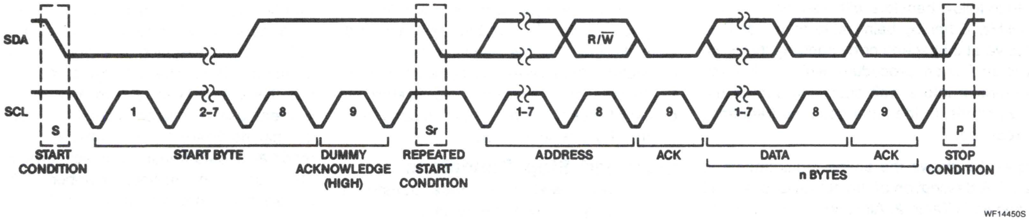

SDA
SCL
R/W̄
1
2-7
8
9
1-7
8
9
1-7
8
9
S
Sr
P
START CONDITION
START BYTE
DUMMY ACKNOWLEDGE (HIGH)
REPEATED START CONDITION
ADDRESS
ACK
DATA
ACK
n BYTES
STOP CONDITION
WF14450S

Figure 22. Data Transfer Low Speed Mode

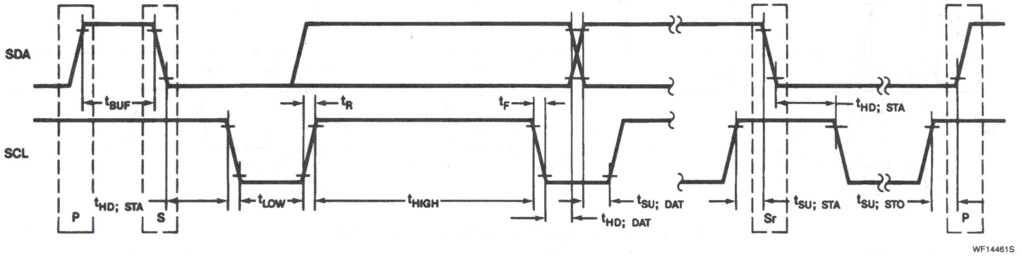

SDA
SCL
tBUF
tR
tF
tHD; STA
P
tHD; STA
S
tLOW
tHIGH
tSU; DAT
tHD; DAT
Sr
tSU; STA
tSU; STO
P
WF14461S

Figure 23. Timing Low Speed Mode

LOW SPEED MODE

CLOCK : $t_{Low} = 130\mu s \pm 25\mu s$
DUTY CYCLE : $t_{High} = 390\mu s \pm 25\mu s$
: 1:3 Low-to-High (Duty cycle of clock generator)
START BYTE : 0000 0001
MAX. NO. OF BYTES : UNRESTRICTED
PREMATURE TERMINATION OF TRANSFER : NOT ALLowED
ACKNOWLEDGE CLOCK BIT : ALWAYS PROVIDED
ACKNOWLEDGEMENT OF SLAVES : OBLIGATORY

In this mode, a transfer cannot be terminated during the transmission of a byte.

The bus is considered busy after the first start condition. It is considered free again one minimum clock Low period, 105μs, after the detection of the stop condition. Figure 23 shows the timing requirements in detail, Table 3 explains the abbreviations.

Table 3. Timing Low Speed Mode

SYMBOL	PARAMETER	LIMITS		UNIT
		Min	Max	
t_{BUF}	Time the bus must be free before a new transmission can start	105		μs
$t_{HD;\ STA}$	Hold time start condition. After this period the first clock pulse is generated	365		μs
$t_{HD;\ STA}$	Hold time (repeated start condition only)	210		μs
t_{Low}	The Low period of the clock	105	155	μs
t_{High}	The High period of the clock	365	415	μs
$t_{SU;\ STA}$	Setup time for start condition (Only relevant for a repeated start condition)	105	155	μs
t_{HD}; t_{DAT}	Hold time DATA for CBUS compatible masters for I^{2}C devices	 5 0*		 μs μs
$t_{SU;\ DAT}$	Setup time DATA	250		ns
t_R	Rise time of both SDA and SCL lines		1	μs
t_F	Fall time of both SDA and SCL lines		300	ns
$t_{SU;\ STO}$	Setup time for stop condition	105	155	μs

NOTES:
All values referenced to V_{IH} and V_{IL} levels.
* Note that a transmitter must internally provide a hold time to bridge the undefined region (300ns max.) of the falling edge of SCL.

I^{2}C Bus Specification

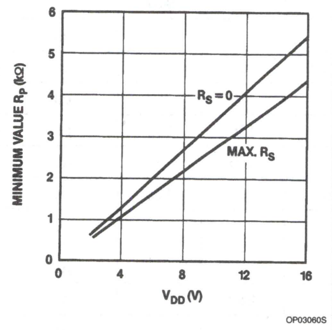
APPENDIX A

Maximum and minimum values of the pull-up resistors R_P and series resistors R_S (See Figure 20).

In a I^{2}C bus system these values depend on the following parameters:

- Supply voltage
- Bus capacitance
- Number of devices (input current + leakage current)

1) The supply voltage limits the minimum value of the R_P resistor due to the specified 3mA as minimum sink current of the output stages, at 0.4V as maximum low voltage. In Graph 1, V_{DD} against R_{Pmin} is shown.

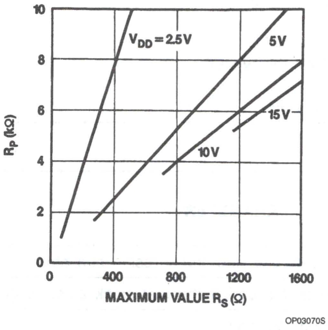


Graph 1

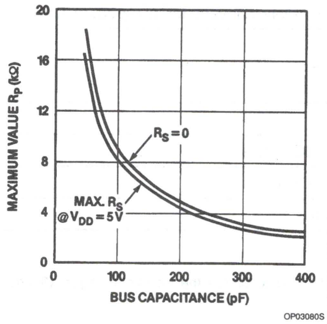
The desired noise margin of 0.1 V_{DD} for the low level limits the maximum value of R_S.

In Graph 2, R_{Smax} against R_P is shown.

2) The bus capacitance is the total capacitance of wire, connections, and pins. This capacitance limits the maximum value of R_P because of the specified rise time of 1μs.



Graph 2

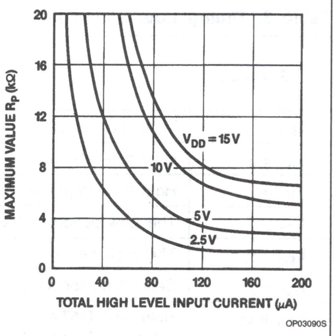


Graph 3

In Graph 3, the bus capacitance – R_{Pmax} relationship is shown.

3) The maximum high-level input current of each input/output connection has a specified value of 10μA max. Due to the desired noise margin of 0.2 V_{DD} for the high level, this input current limits the maximum value of R_P. This limit is dependent on V_{DD}.

In Graph 4 the total high-level input current – R_{Pmax} relationship is shown.



Graph 4

I^{2}C LICENSE

Purchase of Signetics or Philips I^{2}C components conveys a license under the Philips I^{2}C patent rights to use these components in an I^{2}C system, provided that the system conforms to the I^{2}C standard specification as defined by Philips.

Signetics

AN408
SC83C451 Operation of Port 6

Application Note

Microprocessor Products

INTRODUCTION

The features of the 83C451 are shared with the 80C51 or are conventional except for the operation of port 6. The flexibility of this port facilitates high-speed parallel data communications.

This applications note discusses the use of port 6 and is divided into the following sections:

1. Port 6 as a Processor Bus Interface.
2. Using Port 6 as a Standard Pseudo Bidirectional I/O Port.
3. Implementations of Parallel Printer Ports.

PORT 6 AS A PROCESSOR BUS INTERFACE

Port 6 allows use of the 83C451 as an element on a microprocessor type bus. The host processor could be a general purpose MPU or the data bus of a microcontroller like the 83C451 itself.

This feature allows single or multiple 83C451 controllers to be used on a bus as flexible peripheral processing elements. Applications might include keyboard scanners, serial I/O controllers, servo controllers, etc.

OPERATION

On reset, port 6 is programmed correctly for use as a bus interface (see Figure 2). This prevents the interface from disrupting data on the bus of the host processor during power-up. Software initialization of the CSR (Control Status Register) is not required. A dummy read of port 6 may be required to clear the IBF (Input Buffer Full) flag since it could be set by turn on transients on the bus of the host processor. On reset the CSR of the 83C451 is programmed to allow the following:

1. AFLAG is an input controlling the port select function. If AFLAG is high, the contents of the CSR is output on port 6 when the port is read by the host. If AFLAG is low, then the contents of the output latch is output when port 6 is read by the host.
2. BFLAG is an input controlling the port enable function. In this mode when BFLAG is high, the input latch and the output drivers are disabled and the flags are not affected by the $\overline{\text{IDS}}$ (Input Data Strobe) or $\overline{\text{ODS}}$ (Output Data Strobe) signals. When BFLAG is low, the port is enabled for reading and writing under the control of $\overline{\text{IDS}}$ and $\overline{\text{ODS}}$ pins.

Figure 1 shows one possible example of an 83C451 on a memory bus. This arrangement allows the main processor to query port 6 for flag status without interrupting the 83C451. If the address decoder, shown in Figure 1, enables port 6 on the 83C451 when the address is 8000H or 8001H, and the address line A0 controls the port select feature, then the host processor can read and write to port 6 using address 8000H. Since the port select function is being controlled by the address line A0, the CSR contents can be read by the host processor at address 8001H.

By testing the CSR contents in this way, the host processor can tell if new data has been written to the port 6 output latch since it last read the port or if the 83C451 has read the last byte that the host wrote to the port. Conversely the 83C451 can poll the flags in its CSR to see if the host processor has written to or read from port 6 since the last time it serviced the port.

If desired, an interrupt source for the 83C451 can be derived easily from the port enable source as shown by the dashed line in Figure 1.

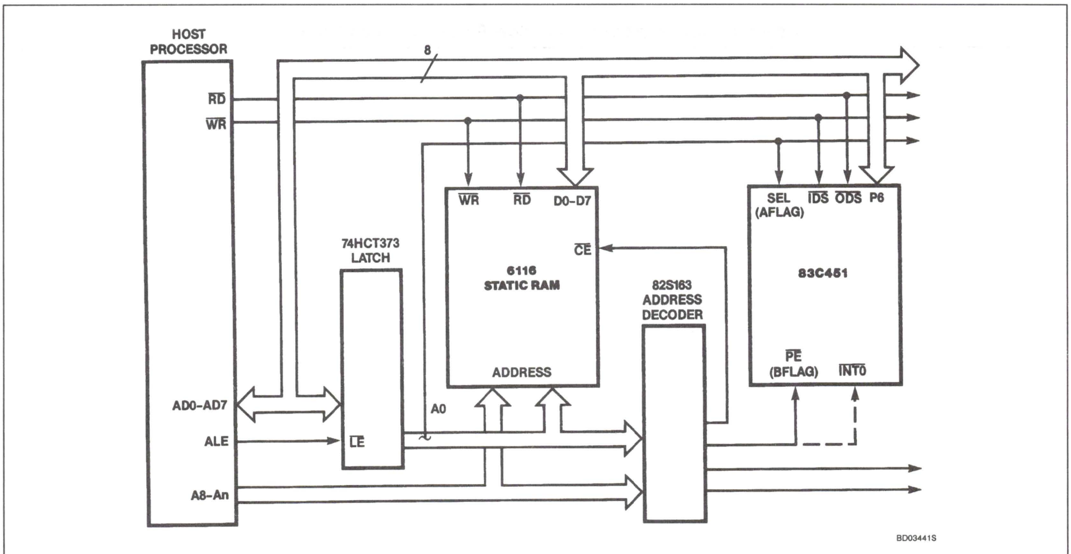


Figure 1. An 83C451 on a Microprocessor Memory Bus

SC83C451 Operation of Port 6

SOFTWARE EXAMPLES

To write to port 6 on the bus shown in Figure 1, the host processor first reads the CSR contents at address 8001H, and tests the input buffer full flag (CSR bit 0). If the flag is clear the host writes a byte to address 8000H. This loads the input buffer latch of port 6 and sets the input buffer full flag.

Conversely the 83C451 polls the IBF flag and reads a byte from port 6 when it finds the flag set. The flag is automatically reset when this internal read occurs.

83C451 ROUTINE TO READ ONE BYTE FROM HOST VIA PORT 6

```
RCVR:    JNB CSR.0,RCVR         ;TEST IBF FLAG
         MOV A,P6               ;WHEN FLAG IS SET READ BYTE
         RET
```

80C51 ROUTINE TO WRITE ONE BYTE TO THE 83C451 PORT 6

If the host processor is an 80C51 the following routine will write a byte of data to the 83C451. The data involved is passed to the routine through register 1.

```
XMIT:    MOV DPTR,8001H
TEST:    MOVX A,@DPTR           ;READ THE CSR
         JB ACC.0,TEST          ;TEST IBF FLAG
         MOV DPTR,8000H
         MOV A,R1
         MOVX @DPTR,A           ;WRITE DATA TO THE 451
         RET
```

83C451 ROUTINE TO WRITE ONE BYTE TO HOST VIA PORT 6

Routines for data transfer in the opposite direction are similar to the above two. The 83C451 version is given below.

```
XMIT:    JB CSR.1,XMIT          ;TEST OBF FLAG
         MOV P6,A               ;WRITE DATA
         RET
```

CSR 7	CSR 6	CSR 5	CSR 4	CSR 3	CSR 2	CSR 1	CSR 0
MB1	MB0	MA1	MA2	OBFC	IDSM	OBF	IBF
1	1	1	1	1	1		

Figure 2. CSR Programmed to Allow Port 6 as a Bus Interface

USING PORT 6 AS A STANDARD QUASI-BIDIRECTIONAL I/O PORT

To use port 6 as a common I/O port, all of the control pins are tied to ground (see Figure 3). On hardware reset, bits 2 – 7 in the CSR are set to one. Port operation and electrical characteristics become identical to port 1 on the 80C51 and the 83C451 ports 1, 4, and 5. No software initialization is required.

If desired, AFLAG and BFLAG can be used as outputs while port 6 is operating as a standard quasi-bidirectional I/O port (see Figure 4). In this case, only $\overline{\text{IDS}}$ and $\overline{\text{ODS}}$ are tied to ground and the CSR is initialized to allow operation of AFLAG and BFLAG as simple outputs (see Figure 5).

IMPLEMENTATION OF PARALLEL PRINTER PORTS USING PORT 6

The 83C451 is an excellent choice for a printer controller. The 83C451 has the facilities to permit all of the intelligent features of a common printer to be handled by a single chip:

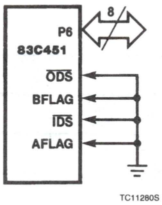


Figure 3. Standard I/O Port on Reset

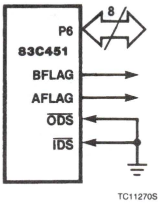


Figure 4. Standard I/O Port on Reset with AFLAG and BFLAG as Outputs

1. The features of port 6 allow a parallel printer port to be designed with only line driving and receiving chips required as additional hardware.
2. The onboard UART allows RS232 interfacing with only level shifting chips added.
3. The 8-bit parallel ports 0 to 6 are ample to drive onboard control functions, even when ports are used for external memory access, interrupts and other functions.
4. The RAM addressing ability of ports 0 and 2 can be used to address up to 64K bytes of a hardware buffer/spooler. AFLAG and BFLAG as simple outputs (see Figure 5).
5. The 64K byte ROM addressing capability allows space for the most sophisticated software.

In addition, either end of a parallel interface can be implemented using port 6, and the interfaces can be interrupt driven or polled in either case.

CSR 7	CSR 6	CSR 5	CSR 4	CSR 3	CSR 2	CSR 1	CSR 0
MB1	MB0	MA1	MA2	OBFC	IDSM	OBF	IBF
0	X	0	X	X	1		

Figure 5. CSR Programmed to Allow AFLAG and BFLAG to Operate as Outputs and Port 6 as a Standard I/O Port

DATA TRANSFER SIGNAL PINS		
Pin No.	Ground Return Pin No.	Signal
1	19	$\overline{\text{STROBE}}$
2	20	DATA 1
3	21	DATA 2
4	22	DATA 3
5	23	DATA 4
6	24	DATA 5
7	25	DATA 6
8	26	DATA 7
9	27	DATA 8
10	28	$\overline{\text{ACKNLG}}$
11	29	BUSY

TYPICAL AUXILIARY PIN FUNCTIONS	
Pin No.	Signal
12	PAPER OUT
14	$\overline{\text{AUTO LINE FEED}}$
16	LOGIC GROUND
17	CHASSIS GND
30	GROUND RETURN
31	$\overline{\text{RESET PRINTER}}$
32	$\overline{\text{ERROR}}$
33	GROUND RETURN
36	$\overline{\text{SLCT IN}}$

Figure 6. Parallel Printer Interface Pin Functions

SC83C451 Operation of Port 6 AN408

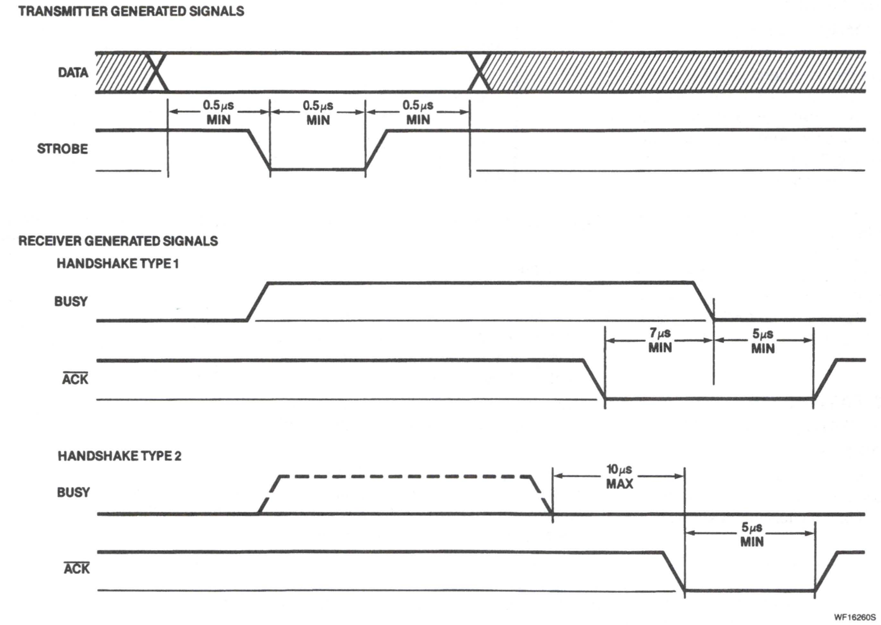

TRANSMITTER GENERATED SIGNALS
DATA
STROBE
0.5μs MIN
0.5μs MIN
0.5μs MIN
RECEIVER GENERATED SIGNALS
HANDSHAKE TYPE 1
BUSY
ACK
7μs MIN
5μs MIN
HANDSHAKE TYPE 2
BUSY
10μs MAX
ACK
5μs MIN
WF16260S

Figure 7. Parallel Printer Interface Signals

THE INTERFACE

Data transfer on a parallel printer interface occurs across eleven signal lines. The other conductors on the standard plug are used as ground returns or for auxiliary functions (see Figure 6). Only the data transfer signals will be considered.

The Data Transfer Format

The parallel printer interfaces are far more standardized in features than their serial counterpart. However at least three significant variations exist in handshake style in printers using generic parallel interfaces. This fact influences the design of both port hardware and software. A good transmitter should be able to drive devices with all three styles of handshakes and a good receiver should generate the handshake most likely compatible with any transmitter.

The Variations

Type 1 — Figure 7 shows a common style of handshake and is the style that will be implemented in the receiver examples. A busy signal and an acknowledge strobe pulse are generated for every byte received.

Type 2 — Another style of handshake generates a busy signal only when the printer will not be able to accept more data for a relatively long time. Acknowledge pulses are created after every byte received. When the busy signal is generated after a byte is received, the associated acknowledge pulse does not occur until **after** the busy signal returns to logic zero. (see Figure 7).

Type 3 — A third handshake style does not generate acknowledge pulses, but a busy signal is produced after every byte is received.

SC83C451 Operation of Port 6 AN408

PARALLEL PRINTER INTERFACES USING POLLING

Transmitter Operation

This application illustrates the flexibility of the port 6 logic in solving an applications problem. We need to be able to handle all types of acknowledge signals that might be received by the transmitter. We will use the $\overline{\text{ODS}}$ pin and output buffer full flag logic to record the receipt of the acknowledge pulse (see Figure 8), but not all parallel receivers generate acknowledge pulses. We could poll the busy signal line, but not all receivers generate busy signals for each byte received; so lack of a busy signal does not imply that we can send another byte. We can, however, expect an acknowledge pulse very shortly after the end of a busy signal if one is going to arrive at all. So we can send a new data byte after having received either a positive transition on the acknowledge line, or shortly after receiving a negative edge on the busy line.

The CSR is programmed to the output only mode. In this mode the $\overline{\text{ODS}}$ pin does not control the output drivers but only the output buffer full flag. The flag serves to record the positive transition of the acknowledge signal. The input latch is not used, but the $\overline{\text{IDS}}$ pin is used to set the input buffer full flag. This is used to record the negative transition at the end of the busy signal. Dummy reads by the 83C451 of port 6 will be used to clear the flag. In this example, the AFLAG mode is set only to place the port in the output only mode. The AFLAG pin is not actually used (see Figure 10).

The transmitter's CSR (control status register) is programmed to the following mode (see Figure 9):

1. CSR bit 6 controls the BFLAG output and therefore the strobe line.
2. The OBF (output buffer full) flag controls the AFLAG* output.
3. The OBF is cleared on the positive edge of the $\overline{\text{ODS}}$ input.
4. The IBF flag is cleared on the negative edge of the $\overline{\text{IDS}}$ strobe.

NOTE:
With this combination of modes set, port 6 is in the output only mode.

Receiver Operation

In receiver operation, the $\overline{\text{IDS}}$ input is used to latch in the data transmitted on receipt of the strobe pulse. The receiver's CSR is programmed to allow the following:

1. The input buffer full flag is output through the BFLAG pin and is used as the busy signal to the transmitter.
2. The IBF flag is set and data is latched on the positive edge of $\overline{\text{IDS}}$.
3. Writing to the CSR bit 4 controls the AFLAG output and therefore the acknowledge line.

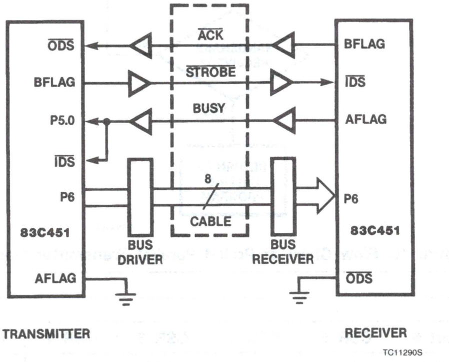


Figure 8. Interconnection for a Parallel Interface Using Polling

CSR 7	CSR 6	CSR 5	CSR 4	CSR 3	CSR 2	CSR 1	CSR 0
MB1	MB0	MA1	MA2	OBFC	IDSM	OBF	IBF
0	1	1	0	0	1		

Figure 9. CSR Programmed for Polled Transmitter Operation

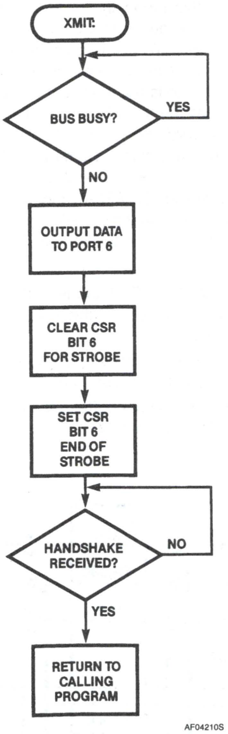

XMIT:
BUS BUSY?
YES
NO
OUTPUT DATA TO PORT 6
CLEAR CSR BIT 6 FOR STROBE
SET CSR BIT 6 END OF STROBE
HANDSHAKE RECEIVED?
NO
YES
RETURN TO CALLING PROGRAM
AF04210S

Figure 10. Flow Chart of Polled Parallel Transmitter Operation

CSR 7	CSR 6	CSR 5	CSR 4	CSR 3	CSR 2	CSR 1	CSR 0
MB1	MB0	MA1	MA2	OBFC	IDSM	OBF	IBF
1	0	0	1	1	1		

Figure 11. CSR Programmed for Polled Parallel Receiver Operation

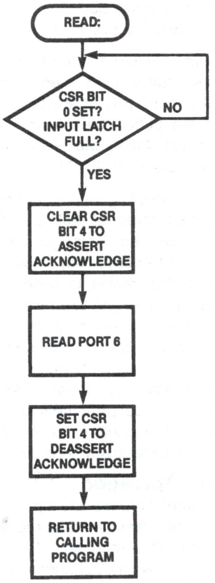

READ:
CSR BIT 0 SET? INPUT LATCH FULL?
NO
YES
CLEAR CSR BIT 4 TO ASSERT ACKNOWLEDGE
READ PORT 6
SET CSR BIT 4 TO DEASSERT ACKNOWLEDGE
RETURN TO CALLING PROGRAM
AF04220S

Figure 12. Flow Chart of Polled Parallel Receiver Operation

SOFTWARE EXAMPLES

This polled parallel transmit routine outputs one byte passed to it in the accumulator.

```
P_INIT:     MOV CSR,#064H         ;INITIALIZE PORT 6 OPERATING MODE
P_OUT:      JB P5.0               ;WAIT IF BUSY SIGNAL IS HIGH
            MOV P6,ACC            ;OUTPUT DATA
            MOV R1,P6             ;DUMMY READ TO CLEAR IBF FLAG
            MOV R1,#02H           ;INITIALIZE DELAY COUNTER
            CLEAR CSR.6           ;START STROBE PULSE
            DJNZ R1,$             ;TIME 6 MICROSECOND STROBE PULSE
            SETB CSR.6            ;END STROBE PULSE
WAIT:       JNB CSR.1,OUT         ;EXIT IF ACKNOWLEDGE RCV'D
            JNB CSR.0,WAIT        ;EXIT IF NEGATIVE BUSY EDGE RCV'D
            RET
```

This polled parallel receive routine places one byte in the accumulator each time it is called.

```
P_INIT:     MOV CSR,#09CH         ;INITIALIZE PORT 6 OPERATING MODE
            MOV R7,P6             ;DUMMY READ TO CLEAR IBF FLAG
P_IN:       JNB CSR.0             ;INPUT BUFFER LATCH FULL?
            CLR CSR.4             ;BEGIN ACKNOWLEDGE PULSE
            MOV R7,#02H           ;INITIALIZE DELAY COUNTER
            DJNZ R7,$             ;TIME ACKNOWLEDGE PULSE
            MOV A,P6              ;READ BYTE - CLEAR BUSY SIGNAL
            MOV R7,#02H           ;INITIALIZE DELAY COUNTER
            DJNZ R7,$             ;TIME ACKNOWLEDGE PULSE
            SETB CSR.4            ;END ACKNOWLEDGE PULSE
            RET
```

INTERRUPT DRIVEN PARALLEL PRINTER INTERFACE

Transmitter Operation

The transmitter's CSR (control status register) is programmed to the following mode (see Figure 14):

1. CSR bit 6 controls the BFLAG output and therefore the strobe line.
2. The OBF (output buffer full) flag controls the AFLAG* output.
3. The OBF is cleared on the positive edge of the $\overline{\text{ODS}}$ (output data strobe) input.
4. The IBF flag is set on the negative edge of the $\overline{\text{IDS}}$ (input data strobe) pin.

NOTE:

With this combination of AFLAG and BFLAG modes set, port 6 is in the output only mode. The output drivers are always enabled and the $\overline{\text{ODS}}$ input is only used to clear the OBF flag.

INT0 is programmed to be negative edge sensitive and is connected to the OBF flag through the AFLAG pin. The OBF is cleared on the positive edge of $\overline{\text{ODS}}$. The net result is that INT0 is triggered on the end of the ACK pulse (a positive edge). This signals the transmitter that another byte may be transmitted. The transmitting 83C451 is free to do other tasks prior to this interrupt.

In this routine, Figure 15, the main program establishes a buffer in data memory ended by an ASCII end of text character. To begin outputting the buffer the routine PSEND is called. The rest of the buffer is emptied by the interrupt vectors to PSEND1.

For printers which generate acknowledge pulses, output rates of 25k transfers per second are achieved. Timer generated interrupts are used to periodically return program execution to the routine to service non-acknowledging printers and to provide a timeout feature. Non acknowledging printers are serviced at a rate of about 2.5k transfers per second. This maximum rate may be varied by adjusting the timer reload value. As written, the time out procedure attempts to retransmit a byte when the printer has not acknowledged for an excessively long time.

Receiver Operation

In receiver operation, the $\overline{\text{IDS}}$ input is used to latch in the data transmitted on receipt of the strobe pulse. The receiver's CSR is programmed to allow the following (see Figure 16):

1. The input buffer full flag is output through the BFLAG pin and is used as the busy signal to the transmitter. The IBF flag is set and data is latched on the positive edge of $\overline{\text{IDS}}$.
2. Writing to the CSR bit 4 controls the AFLAG output and therefore the acknowledge line.

The receiver is interrupted on the negative edge of the data strobe. Data is latched in on the positive edge of the strobe pulse (see Figure 17). Since the strobe pulse is normally very short there is little time lost between receiving the interrupt and having valid data in the input latch. The receiver is free to do other tasks prior to receiving the $\overline{\text{INT0}}$ interrupt.

SOFTWARE EXAMPLES

The software for the interrupt driven parallel receiver is similar to the polled receiver example. However, after an interrupt is received, this routine checks to confirm that data has been latched by the positive edge of the strobe pulse before proceeding with the routine.

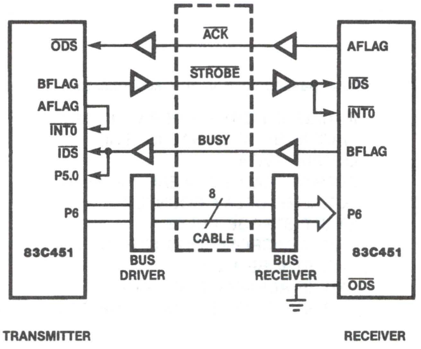


Figure 13. Interrupt Driven Parallel Interfaces Using 83C451 Controllers

CSR 7	CSR 6	CSR 5	CSR 4	CSR 3	CSR 2	CSR 1	CSR 0
MB1	MB0	MA1	MA2	OBFC	IDSM	OBF	IBF
0	1	1	0	0	1		

Figure 14. CSR Programming for Use as an Interrupt Driven Parallel Transmitter

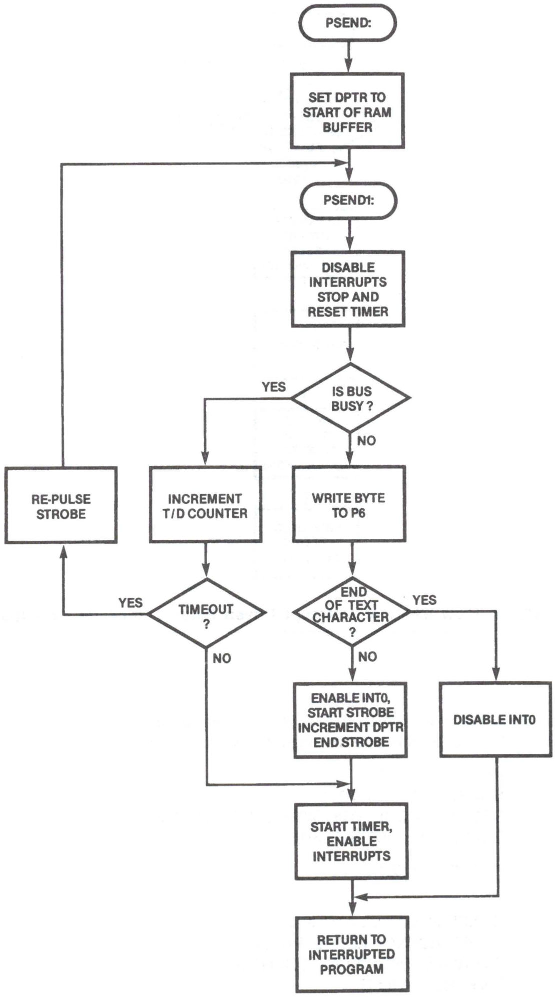

PSEND:
SET DPTR TO START OF RAM BUFFER
PSEND1:
DISABLE INTERRUPTS STOP AND RESET TIMER
IS BUS BUSY ?
YES
NO
RE-PULSE STROBE
INCREMENT T/D COUNTER
WRITE BYTE TO P6
YES
TIMEOUT ?
NO
END OF TEXT CHARACTER ?
YES
NO
ENABLE INT0, START STROBE INCREMENT DPTR END STROBE
DISABLE INT0
START TIMER, ENABLE INTERRUPTS
RETURN TO INTERRUPTED PROGRAM
AF04230S

Figure 15. Flow Chart for an Interrupt Driven Parallel Transmitter

CSR 7	CSR 6	CSR 5	CSR 4	CSR 3	CSR 2	CSR 1	CSR 0
MB1	MB0	MA1	MA2	OBFC	IDSM	OBF	IBF
1	0	0	1	1	0		

Figure 16. CSR Programming for Use as an Interrupt Driven Parallel Receiver

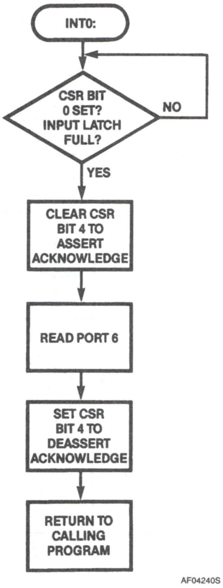

INT0:
CSR BIT 0 SET? INPUT LATCH FULL?
NO
YES
CLEAR CSR BIT 4 TO ASSERT ACKNOWLEDGE
READ PORT 6
SET CSR BIT 4 TO DEASSERT ACKNOWLEDGE
RETURN TO CALLING PROGRAM
AF04240S

Figure 17. Flow Chart of Interrupt Driven Parallel Receiver Operation

SOFTWARE EXAMPLES

The software for the interrupt driven parallel receiver is similar to the polled receiver example. However, after an interrupt is received, this routine checks to confirm that data has been latched by the positive edge of the strobe pulse before proceeding with the routine.

```
INIT:     MOV CSR,#090H      ;INITIALIZE CSR
          SETB EX0           ;ENABLE INTERRUPT 0
          SETB IT0           ;SET NEG EDGE TRIGGERED INTERRUPTS
          SETB EA            ;ENABLE ALL INTERRUPTS

ORG EXTI0                    ;INTERRUPT 0 VECTOR
          JMP RCVR

RCVR:     JNB CSR.0,$        ;CONFIRM DATA LATCHED
          CLR CSR.4          ;START ACKNOWLEDGE PULSE
          MOV R7,#02H        ;INITIALIZE THE DELAY COUNTER
          DJNZ R7,$          ;TIME ACK PULSE
          MOV A,P6           ;READ BYTE - RESET BUSY LINE
          MOV R7,#02H        ;INITIALIZE THE DELAY COUNTER
          DJNZ R7,$          ;TIME ACK PULSE
          SETB CSR.4         ;END ACK PULSE
          RETI
```

This is the software for the interrupt driven parallel transmitter example.

```
; XMIT ROUTINE DRIVEN BY ACK PULSE GENERATED INTERRUPTS, OR TIMER GENERATED INTERRUPTS
; FOR NON ACKNOWLEDGING PRINTERS. READS DATA BUFFER IN EXTERNAL RAM STARTING AT 100H
; AND READING UNTIL 04H IS FOUND.

    ORG RESET
     JMP 26H
    ORG TIMER0
     JMP PSEND1
    ORG EXTI0
     JMP PSEND1
    ORG 26H
          MOV CSR,#064H             ;PORT 6 MODE
          MOV TMOD,#002H            ;CONFIGURE TIMER 0 TO 16 BITS
          SETB TO0                  ;INT0 IS EDGE TRIGGERED
          SETB EA                   ;ENABLE INTERRUPTS
PSEND:    MOV DPTR,#0100H           ;SET DPTR TO START OF TEXT
                                    ;BUFFER
PSEND1:   CLR EA                    ;DISABLE INTERRUPTS AND STOP
                                    ;TIMER
          CLR TR0                   ;IF ENABLED
          CLR ET0
          MOV R7,00H                ;CLEAR TIMEOUT COUNTER
          MOV R6,00H
          MOV TH0,# - 4             ;SET TIMER INTERRUPT PERIOD
          MOV TL0,#00H
          JB 0C8H,BB                ;BUS BUSY
          MOV ACC,#00H              ;CLEAR ACCUMULATOR
          MOVX A,@DPTR              ;RETRIEVE FIRST BYTE
          MOV P6,ACC                ;OUTPUT FIRST BYTE
          CJNE A,#004H, CONT1       ;LOOK FOR END OF TEXT
          JMP EOTB
CONT1:    SETB EX0                  ;ENABLE INT0
          CLR 0EEH                  ;START STROBE PULSE
          INC DPTR
          MOV ACC,DPH               ;LOOK FOR PHYSICAL END OF
          JB ACC.2,EOTB             ;TEXT BUFFER
          SETB 0EEH                 ;END STROBE PULSE
          JMP CONT
EOTB:     CLR EX0                   ;END OF TEXT FOUND, DISABLE
                                    ;INT0
          SETB 0EEH
          SETB EA
          RETI
BB:       INC R7                    ;COUNT TIMER TIMEOUTS ON
                                    ;BUS BUSY
          CJNE R7,#00H, CONT        ;LOOK FOR OVERFLOW
          INC R6                    ;COUNT OVERFLOWS
          CJNE R6,#10H, CONT        ;TIMEOUT APPROX 5 SEC
          JMP TO
CONT:     SETB TR0                  ;ENABLE TIMER INTERRUPT
          SETB ET0                  ;START TIMER
          SETB EA
          RETI
TO:       CLR 0C9H                  ;SEND NEW STROBE PULSE IN
                                    ;RESPONSE TO TIMEOUT
          NOP
          NOP
          MOV R6,#00H               ;RESET TO COUNTER
          MOV R7,#00H
          SETB 0C9H                 ;END OF STROBE PULSE
          JMP PSEND1
```

Signetics

Microprocessor Products

8X305
Microcontroller

Product Specification

FEATURES

- **Fetch, Decode, and Execute a 16-bit instruction in a minimum of 200 ns (one machine cycle)**
- **Bit-oriented instruction set (addressable single-or-multiple bit subfields)**
- **Separate buses for Instruction, Instruction Address and Three-State I/O**
- **Thirteen 8-bit general-purpose working registers**
- **Source/destination architecture**
- **Bipolar low-power Schottky technology/TTL inputs and outputs**
- **On-chip oscillator and timing generation**
- **Single +5V supply**
- **0.9-in. 50-pin DIP**
- **68-Pin PLCC**

PRODUCT DESCRIPTION

The Signetics 8X305 Microcontroller (Figure 1) is a high-speed bipolar microprocessor implemented with low-power Schottky technology. In a single chip, the 8X305 combines speed, flexibility, and a bit-oriented instruction set. These features and other basic characteristics of the chip combine to provide cost-effective solutions for a broad range of applications. The 8X305 is particularly useful in systems that require high-speed bit manipulations — sophisticated controllers, data communications, very fast interface control, and other applications of a similar nature.

The 8X305 can fetch, decode, and execute a 16-bit instruction word in a minimum of 200ns. Within one instruction cycle, the 8-bit data-processing path can be programmed to rotate, mask, shift, and/or merge single or multiple bit subfields and, in addition, perform an ALU operation. In the same instruction, an external data field can be input, processed, and output to a specified destination — likewise, single or multiple bit data fields can be internally moved from a given source to a given destination. To summarize, fixed or variable-length data fields can be fetched, processed, operated on by the ALU, and moved to a different location — all in a timeframe of 200ns. To interface with I/O and program memory, the 8X305 uses a 13-bit instruction address bus, a 16-bit instruction bus, an 8-bit bidirectional multiplexed I/O data/address bus and a 5-bit I/O control bus.

A wide selection of I/O devices, interface chips, and special-purpose parts are available for systems use. In most applications, the more powerful 8X305 is functionally interchangeable with its predecessor — the 8X300.

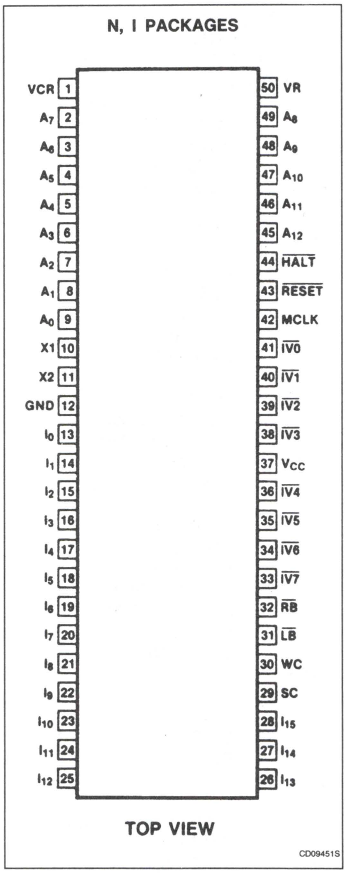
ASSOCIATED DOCUMENTATION

Other documents directly relating to *design* and *applications use* of the 8X305 Microcontroller are:

- Product Capabilities Manual
- 8X305 Users Manual

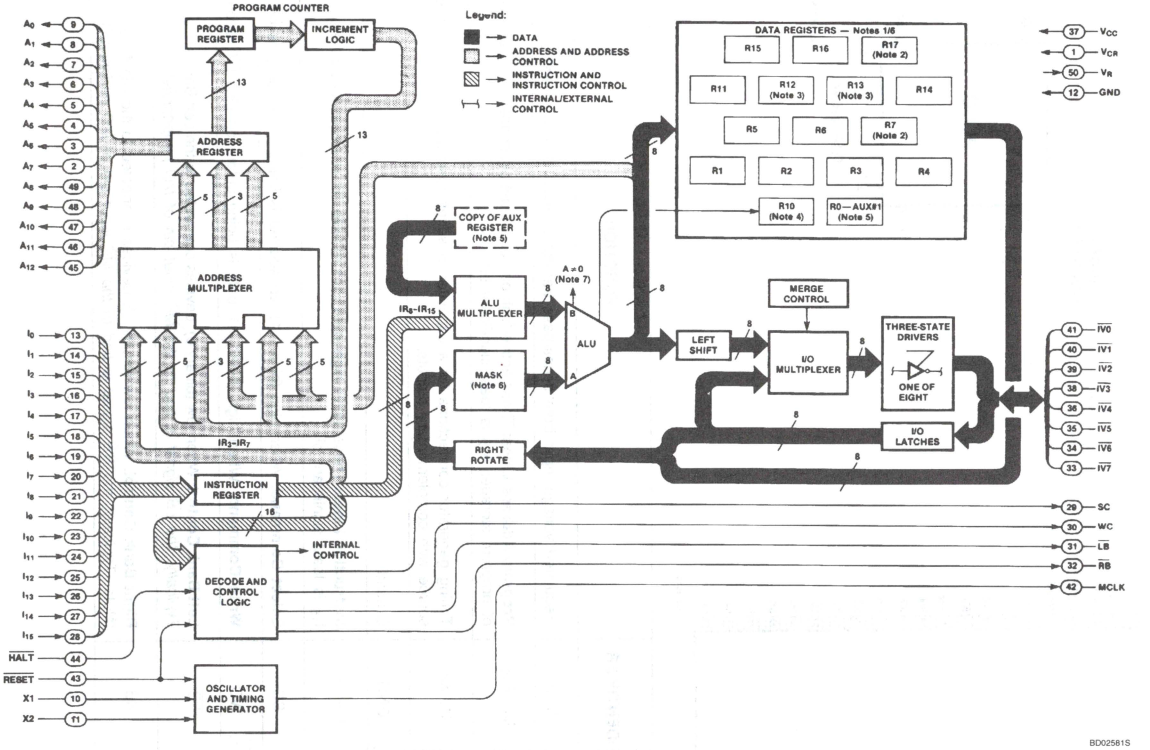
These documents and other current literature (Data Sheets, Product Bulletins, Applications Notes, etc.) are available at all Signetics Sales and Service Offices — see rear cover of this data sheet for the office in your locality.

PIN CONFIGURATION


N, I PACKAGES
VCR 1
A_7 2
A_6 3
A_5 4
A_4 5
A_3 6
A_2 7
A_1 8
A_0 9
X1 10
X2 11
GND 12
I_0 13
I_1 14
I_2 15
I_3 16
I_4 17
I_5 18
I_6 19
I_7 20
I_8 21
I_9 22
I_{10} 23
I_{11} 24
I_{12} 25
50 VR
49 A_8
48 A_9
47 A_{10}
46 A_{11}
45 A_{12}
44 $\overline{HALT}$
43 $\overline{RESET}$
42 MCLK
41 $\overline{IV0}$
40 $\overline{IV1}$
39 $\overline{IV2}$
38 $\overline{IV3}$
37 V_{CC}
36 $\overline{IV4}$
35 $\overline{IV5}$
34 $\overline{IV6}$
33 $\overline{IV7}$
32 $\overline{RB}$
31 $\overline{LB}$
30 WC
29 SC
28 I_{15}
27 I_{14}
26 I_{13}
TOP VIEW
CD09451S

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
50-Pin plastic DIP	N8X305N
50-Pin ceramic DIP	N8X305I
68-Pin PLCC	N8X305A


PROGRAM COUNTER
PROGRAM REGISTER
INCREMENT LOGIC
ADDRESS REGISTER
ADDRESS MULTIPLEXER
INSTRUCTION REGISTER
DECODE AND CONTROL LOGIC
INTERNAL CONTROL
OSCILLATOR AND TIMING GENERATOR
IR₃–IR₇
IR₈–IR₁₅
COPY OF AUX REGISTER (Note 5)
ALU MULTIPLEXER
MASK (Note 6)
RIGHT ROTATE
ALU
A ≠ 0 (Note 7)
A
B
LEFT SHIFT
MERGE CONTROL
I/O MULTIPLEXER
THREE-STATE DRIVERS
ONE OF EIGHT
I/O LATCHES
DATA REGISTERS — Notes 1/6
R15
R16
R17 (Note 2)
R11
R12 (Note 3)
R13 (Note 3)
R14
R5
R6
R7 (Note 2)
R1
R2
R3
R4
R10 (Note 4)
R0—AUX#1 (Note 5)
Legend:
DATA
ADDRESS AND ADDRESS CONTROL
INSTRUCTION AND INSTRUCTION CONTROL
INTERNAL/EXTERNAL CONTROL
13
5
3
16
8
A₀ 9
A₁ 8
A₂ 7
A₃ 6
A₄ 5
A₅ 4
A₆ 3
A₇ 2
A₈ 49
A₉ 48
A₁₀ 47
A₁₁ 46
A₁₂ 45
I₀ 13
I₁ 14
I₂ 15
I₃ 16
I₄ 17
I₅ 18
I₆ 19
I₇ 20
I₈ 21
I₉ 22
I₁₀ 23
I₁₁ 24
I₁₂ 25
I₁₃ 26
I₁₄ 27
I₁₅ 28
HALT 44
RESET 43
X1 10
X2 11
37 V_CC
1 V_CR
50 V_R
12 GND
41 IV0
40 IV1
39 IV2
38 IV3
36 IV4
35 IV5
34 IV6
33 IV7
29 SC
30 WC
31 LB
32 RB
42 MCLK
BD02581S

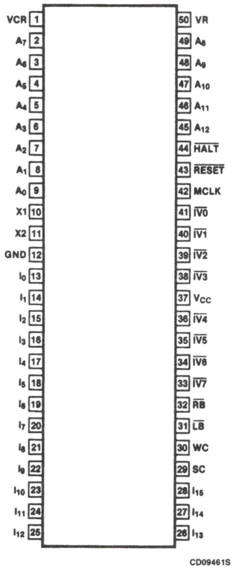
NOTES:

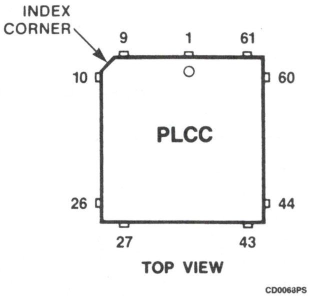
1. Registers R1 – R6, R11 and R14 – R16 are general-purpose working registers.
2. In any instruction where R7 (IVL) or R17 (IVR) is specified as the destination, the 8-bit value is output on the IV bus as an IV device enable address (SC = High) — R7 = left bank and R17 = right bank; the results are also stored into the specified internal register and may later be accessed as source data.
3. R12 and R13 are general-purpose working registers for all operations except transmit (XMIT).
4. The least significant bit of register R10 (OVF) is used to reflect the carryout status resulting from the most recent ADD operation.
5. Auxiliary register R0 #1 is a general-purpose working register that holds the implied operand for Arithmetic and Logical operations; the content of this register is repeated in AUX #2 (shown dotted). The duplicate register is physically part of the ALU and is shown separate only for layout convenience.
6. Internal working registers cannot be operated on by the MASK logic.
7. During NZT instructions the ALU tests for all bits equal to "0" (Transfer if A ≠ 0) — refer to BASIC OPERATIONS that follow.

Figure 1. Architecture and Pin Designations for 8X305 Microcontroller
Pin Numbers Shown are for DIP Packaging

Microcontroller

8X305


VCR 1
A_7 2
A_6 3
A_5 4
A_4 5
A_3 6
A_2 7
A_1 8
A_0 9
X1 10
X2 11
GND 12
I_0 13
I_1 14
I_2 15
I_3 16
I_4 17
I_5 18
I_6 19
I_7 20
I_8 21
I_9 22
I_{10} 23
I_{11} 24
I_{12} 25
50 VR
49 A_8
48 A_9
47 A_{10}
46 A_{11}
45 A_{12}
44 $\overline{\text{HALT}}$
43 $\overline{\text{RESET}}$
42 MCLK
41 $\overline{\text{IV0}}$
40 $\overline{\text{IV1}}$
39 $\overline{\text{IV2}}$
38 $\overline{\text{IV3}}$
37 V_{CC}
36 $\overline{\text{IV4}}$
35 $\overline{\text{IV5}}$
34 $\overline{\text{IV6}}$
33 $\overline{\text{IV7}}$
32 $\overline{\text{RB}}$
31 $\overline{\text{LB}}$
30 WC
29 SC
28 I_{15}
27 I_{14}
26 I_{13}
CD09461S


INDEX CORNER
9
1
61
10
60
PLCC
26
44
27
43
TOP VIEW
CD0068PS

PLCC PIN NO.	DIP PIN NO.	IDENTIFIER	FUNCTION
1, 68	1	VCR	Regulated voltage input from series-pass transistor (2N5320 or equivalent).
4-11, 62-66	2-9, 45-49	$A_0 - A_{12}$	**Program Address Lines:** These active-high outputs permit direct addressing of up to 8192 words of program storage; A_{12} is least significant bit.
12, 13	10, 11	X1, X2	Timing generator connections for a capacitor, a series resonant crystal, or an external clock source with complementary outputs.
2,3, 14-16	12	GND	Ground.
17-23, 28-36	13-28	$I_0 - I_{15}$	**Instruction Lines:** These active-high input lines receive 16-bit instructions from program storage; I_{15} is least significant bit.
37	29	SC	**Select Command:** When high (binary 1), an address is being output on pins $\overline{\text{IV0}}$ through $\overline{\text{IV7}}$.
38	30	WC	**Write Command:** When high (binary 1), data is being output on pins $\overline{\text{IV0}}$ through $\overline{\text{IV7}}$.
39	31	$\overline{\text{LB}}$	**Left Bank Control:** When low (binary 0), devices connected to the Left Bank are accessed. (Note: *Typically, the $\overline{\text{LB}}$ signal is tied to the $\overline{\text{ME}}$ input pin of I/O peripherals*).
45	32	RB	**Right Bank Control:** When low (binary 0), devices connected to the Right Bank are accessed (Note. *Typically, the $\overline{\text{RB}}$ signal is tied to the $\overline{\text{ME}}$ input pin of I/O peripherals*).

PLCC	DIP	IDENTIFIER	FUNCTION
PIN NO.	PIN NO.		
46-49, 55-58	33-36, 38-41	$\overline{\text{IV0}}$ – $\overline{\text{IV7}}$	**Interface Vector** (Input/Output Bus) — these bidirectional active-low three-state lines communicate data and/or addresses to I/O devices and memory locations. A low voltage level equals a binary "1"; $\overline{\text{IV7}}$ is Least Significant Bit.
50-52	37	V_{CC}	+5V power supply.
59	42	MCLK	**Master Clock:** This active-high output signal is used for clocking I/O devices and/or synchronization of external logic.
60	43	$\overline{\text{RESET}}$	When $\overline{\textbf{RESET}}$ input is low (binary 0), the 8X305 is initialized — sets Program Counter/Address Register to zero and inhibits MCLK. For the period of time $\overline{\text{RESET}}$ is low, the Left Bank/Right Bank ($\overline{\text{LB}}/\overline{\text{RB}}$) signals are forced high asynchronously.
61	44	$\overline{\text{HALT}}$	When $\overline{\textbf{HALT}}$ input is low (binary 0), internal operation of the 8X305 stops at the start of next instruction; MCLK is not inhibited nor is any internal register affected. However, both the Left Bank/Right Bank ($\overline{\text{LB}}/\overline{\text{RB}}$) signals are synchronously driven high during the first quarter of the instruction cycle time and remain high during the time HALT is low.
67	50	VR	Internally-generated reference output voltage for external series-pass regulator transistor.
24-27, 40-44, 53, 54	–	No Connect	

NOTE:
Multiple V_{CC}, GND, and V_{CR} pins must be externally connected.

Figure 2. Designations and Descriptions for Pins of 8X305 Microcontroller.

FUNCTIONAL OPERATION

Typical System Configuration

Although the system hookup shown in Figure 3 is of the simplest form, it provides a fundamental look at the 8X305 Microcontroller and peripheral relationships. As indicated, the 8X305 can directly address up to 8K words of program storage — either ROM or PROM. The user interface ($\overline{\text{IV0}}$ through $\overline{\text{IV7}}$) is capable of uniquely addressing 256 Input/Output locations and, with additional bank bits ($\overline{\text{LB}}$, $\overline{\text{RB}}$), this number is expanded to 512 — each bank comprising 256 addressable locations. The addressable locations of each bank can be used in a variety of ways; a simple method of implementation is shown in Figure 3. When $\overline{\text{LB}}$ is active low, the left bank is enabled and any one of 256 locations within the RAM memory can be accessed for input/output operations. A similar set of "enable/access" conditions are applicable to the right bank when $\overline{\text{RB}}$ is active low.

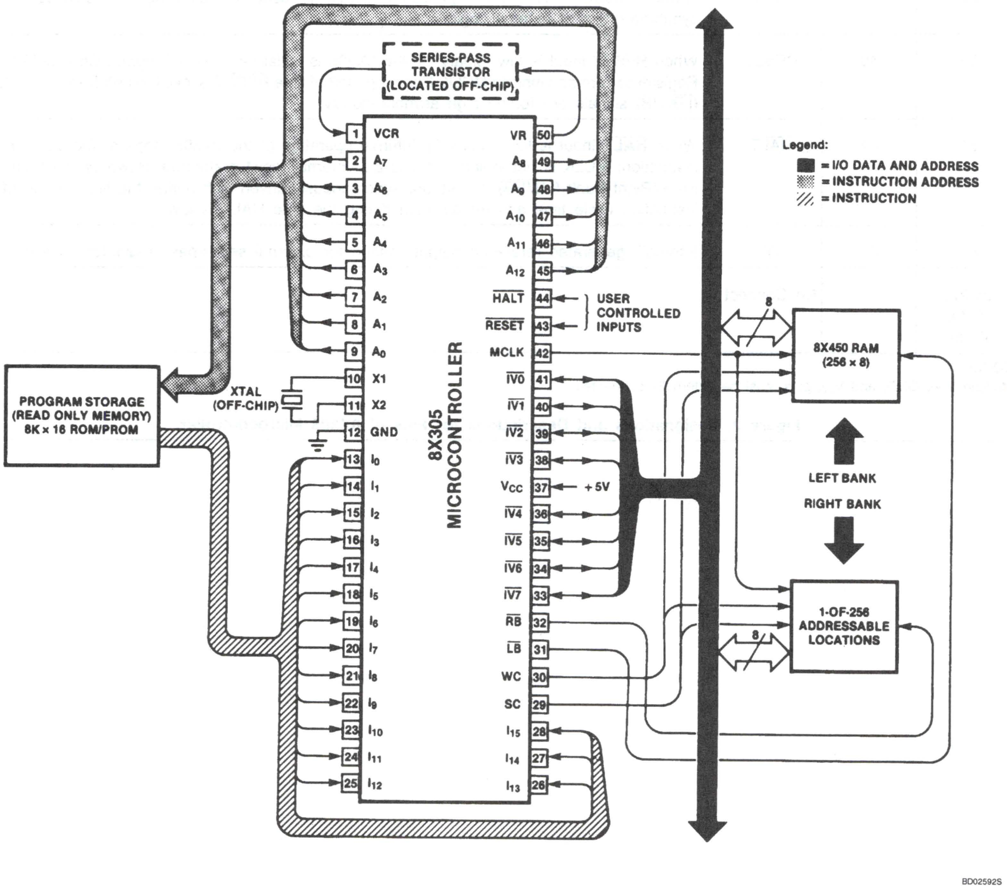

SERIES-PASS TRANSISTOR (LOCATED OFF-CHIP)
1 VCR
2 A7
3 A6
4 A5
5 A4
6 A3
7 A2
8 A1
9 A0
10 X1
11 X2
12 GND
13 I0
14 I1
15 I2
16 I3
17 I4
18 I5
19 I6
20 I7
21 I8
22 I9
23 I10
24 I11
25 I12
50 VR
49 A8
48 A9
47 A10
46 A11
45 A12
44 HALT
43 RESET
42 MCLK
41 IV0
40 IV1
39 IV2
38 IV3
37 VCC
36 IV4
35 IV5
34 IV6
33 IV7
32 RB
31 LB
30 WC
29 SC
28 I15
27 I14
26 I13
8X305 MICROCONTROLLER
USER CONTROLLED INPUTS
+5V
XTAL (OFF-CHIP)
PROGRAM STORAGE (READ ONLY MEMORY) 8K x 16 ROM/PROM
Legend:
■ = I/O DATA AND ADDRESS
▒ = INSTRUCTION ADDRESS
▨ = INSTRUCTION
8
8X450 RAM (256 x 8)
LEFT BANK
RIGHT BANK
8
1-OF-256 ADDRESSABLE LOCATIONS
BD02592S

Figure 3. Typical 8X305 System Hookup

BASIC OPERATIONS OF 8X305

Refer to a later discussion of "Instruction Fields" for a detailed examination of all operand fields and subdivisions thereof — "S" (S_0, S_1), "D" (D_0, D_1), "R", "L", "J", and "A".

MOVE OPERATIONS

REGISTER-TO-REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
R0-R17 as specified by "S" field of instruction.	Right rotate as specified by "R" field of instruction.	No operation.	No operation.	R0-R7, R11-R17 as specified by "D" field of instruction.

REGISTER-TO-$\overline{\text{IV}}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
R0-R17 as specified by "S" field of instruction.	No operation.	No operation.	Shift and merge as specified by "D_0" and "L" fields of instruction.	Variable length field of $\overline{\text{IV}}$ bus — Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) as specified by "D_0" and "D_1" fields of instruction.

$\overline{\text{IV}}$ BUS-TO-REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
Variable length field of $\overline{\text{IV}}$ bus — Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) as specified by "S_1" and "S_0" fields of instruction.	Right rotate and mask as specified by "S_0" and "L" fields of instruction.	No operation.	No operation.	R0-R7, R11-R17 as specified by "D" field of instruction.

$\overline{\text{IV}}$ BUS-TO-$\overline{\text{IV}}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
Variable length field of $\overline{\text{IV}}$ bus — Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) as specified by "S_1" and "S_0" fields of instruction.	Right rotate and mask as specified by "S_0" and "L" fields of instruction.	No operation.	Shift and merge as specified by "D_0" and "L" fields of instruction.	Variable length field of $\overline{\text{IV}}$ bus — Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) as specified by "D_0" and "D_1" fields of instruction.

ADD OPERATIONS

REGISTER-to-REGISTER → SOURCE → PRE-ALU → POST-ALU → DEST
REGISTER-to-$\overline{IV}$ BUS → SOURCE → PRE-ALU → POST-ALU → DEST
$\overline{IV}$ BUS-to-REGISTER → SOURCE → PRE-ALU → POST-ALU → DEST
$\overline{IV}$ BUS-to-$\overline{IV}$ BUS → SOURCE → PRE-ALU → POST-ALU → DEST

Same as MOVE operations, except source data is ADDed to contents of AUXiliary Register R0 via the ALU; if appropriate, Overflow Register R10 (OVF) is also set.

AND OPERATIONS

REGISTER-to-REGISTER → SOURCE → PRE-ALU → POST-ALU → DEST
REGISTER-to-$\overline{IV}$ BUS → SOURCE → PRE-ALU → POST-ALU → DEST
$\overline{IV}$ BUS-to-REGISTER → SOURCE → PRE-ALU → POST-ALU → DEST
$\overline{IV}$ BUS-to-$\overline{IV}$ BUS → SOURCE → PRE-ALU → POST-ALU → DEST

Same as MOVE operations, except source data is ANDed with contents of AUXiliary Register R0 via the ALU.

EXCLUSIVE OR (XOR) OPERATIONS

REGISTER-to-REGISTER → SOURCE → PRE-ALU → POST-ALU → DEST
REGISTER-to-$\overline{IV}$ BUS → SOURCE → PRE-ALU → POST-ALU → DEST
$\overline{IV}$ BUS-to-REGISTER → SOURCE → PRE-ALU → POST-ALU → DEST
$\overline{IV}$ BUS-to-$\overline{IV}$ BUS → SOURCE → PRE-ALU → POST-ALU → DEST

Same as MOVE operations, except source data is Exclusively ORed with contents of AUXiliary Register R0 via the ALU.

EXECUTE (XEC) OPERATIONS

XEC, REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
R0 – R17 as specified by "S" field of instruction.	No operation.	Add source data to 8-bit field specified by instruction literal ($0 \leqslant J \leqslant 377_8$).	No operation.	Replace 8 LSB of Address Register* with 8-bit sum from ALU. *PGM CTR unchanged.

XEC, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
Left or Right Bank of $\overline{IV}$ bus as specified by "S" field of instruction.	Rotate and mask as specified by "S_0" and "L" fields of instruction.	Add masked field of source data to 5-bit literal specified by "J" field of instruction ($0 \leqslant J \leqslant 37_8$).	No operation.	Replace 5 LSB of Address Register* with 5-bit sum from ALU. *PGM CTR unchanged.

NON-ZERO TRANSFER (NZT) OPERATIONS

NZT, REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DESTINATION
R0-R17 as specified by "S" field of instruction.	No operation.	Test contents of source register for all zeroes.	If S = 0, increment PC by 1; if S ≠ 0, replace 8 LSB of AR and PC with literal specified by "J" field of instruction.

NZT, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DESTINATION
Left or Right Bank of $\overline{IV}$ bus as specified by "S" field of instruction.	Rotate and mask as specified by "S_0" and "L" fields of instruction.	Test contents of masked field for all zeroes.	If S = 0, increment PC by 1; if S ≠ 0, replace 5 LSB of AR and PC with literal specified by "J" field of instruction.

TRANSMIT (XMIT) OPERATIONS

XMIT, REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
$0 \leqslant J \leqslant 377_8$ – value specified by "J" field of instruction.	No operation.	No operation.	No operation.	R0-R7, R11, R14-R17. Load 8-bit integer specified by "J" field into register specified by "D" field.

XMIT 8-BIT IMMEDIATE, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
$0 \leqslant J \leqslant 377_8$ – value specified by "J" field of instruction.	No operation.	No operation.	No operation.	Left (R12) or Right (R13) Bank of $\overline{IV}$ bus as specified by "D" field of instruction.

XMIT VARIABLE-BIT FIELD IMMEDIATE $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
$0 \leqslant J \leqslant 37_8$ – value specified by "J" field of instruction.	No operation.	No operation.	Shift and merge source data as specified by "D_0" and "L" fields of instruction.	Left or Right Bank of $\overline{IV}$ bus as specified by "D" field of instruction.

Table 1.

$\overline{\text{LB}}$	$\overline{\text{RB}}$	FUNCTION
Low	Low	This state is not generated by the 8X305.
Low	High	Enable left bank devices.
High	Low	Enable right bank devices.
High	High	Disable all devices; $\overline{\text{IV}}$ bus is 3-State.

Table 2.

$\overline{\text{LB}}/\overline{\text{RB}}$	SC	WC	FUNCTION
High	Low	Low	The $\overline{\text{IV}}$ bus is 3-State and not looking for input data.
Low	Low	Low	The $\overline{\text{IV}}$ bus is reading input data.
Low	Low	High	Data is being output.
Low	High	Low	Address is being output.
X	High	High	This condition is never generated.

Program Storage Interface

As shown in Figure 3, program storage is connected to output address lines A_0 through A_{12} (A_{12} = LSB) and input instruction lines I_0 through I_{15}. An address output on A_0/A_{12} identifies one 16-bit instruction word in program storage. The instruction word is subsequently input on I_0/I_{15} and defines the Micro-Controller operation which is to follow — one instruction word equals one completed operation. Any TTL-compatible memory can be used for program storage provided the worst-case access time is compatible with the instruction cycle time used for the application — see timing section for appropriate calculations.

I/O Interface and Control

An 8-bit bidirectional I/O bus, referred to as the Interface Vector ($\overline{\text{IV}}$) bus, provides a communication link between the Microcontroller and the two banks of I/O devices. The $\overline{\text{LB}}$ (Left Bank) and $\overline{\text{RB}}$ (Right Bank) control signals identify which bank is enabled; when both $\overline{\text{LB}}$ and $\overline{\text{RB}}$ are high (inactive), neither bank is enabled and the $\overline{\text{IV}}$ bus is inactive (three-state). A functional analysis of the Left and Right Bank signals is shown in Table 1.

Both data and I/O address information are multiplexed on the $\overline{\text{IV}}$ bus. The SC (Select Command) and WC (Write Command) signals distinguish between data and I/O address information as shown in Table 2.

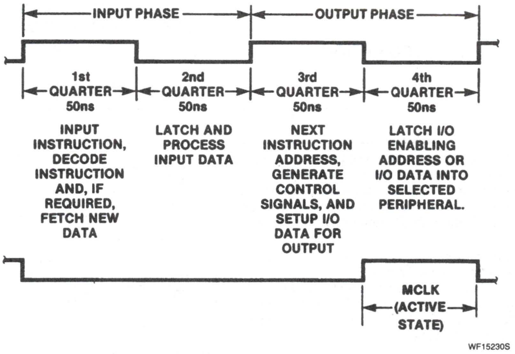
Data Processing

Basically, the data processing path of the 8X305 consists of the Rotate/Mask logic, the Arithmetic Logic Unit (ALU), the Shift/Merge functions, on-chip memory (sixteen 8-bit registers), and the bidirectional $\overline{\text{IV}}$ bus interface with its associated driver circuits and internal latches. The on-board memory and the $\overline{\text{IV}}$ bus are connected to both inputs and outputs of the ALU via internal 8-bit data paths — see Figure 1. Inputs to the ALU are preceded by right-rotate and data-mask functions; the ALU output is followed by the left-shift and merge operations. Depending on the desired operation, any one or all of the functions (Rotate/Mask/Shift/Merge) can operate on 8 bits of data in a single instruction cycle. For a summary of all data-processing capabilities, refer to BASIC OPERATIONS OF THE 8X305 described earlier in this data sheet.

Instruction Cycle

Each operation of the 8X305 is executed in a single instruction cycle. The instruction cycle is internally divided into four equal parts — each part being as short as 50ns. Figure 4 shows the general functions that occur during each quarter cycle; specifics regarding minimum/maximum timing and other critical values are described later in this data sheet. During the first quarter cycle, a new instruction from program storage is input via $I_0 - I_{15}$ and decoded. If an I/O operation is indicated, new data is fetched from a specified internal register or via the $\overline{\text{IV}}$ bus. At the end of the first quarter cycle, the new instruction is latched into the instruction register.

In the second quarter cycle, the I/O input data stabilizes and preliminary processing is completed. At the end of this quarter, the $\overline{\text{IV}}$ latches close and final processing can be accomplished, thus completing the input phase of the instruction cycle. During the third quarter cycle, the address for the next instruction is output to the instruction address bus, $\overline{\text{IV}}$ control signals are generated, and both data and destination are setup for the remainder of the output phase. During the fourth quarter cycle, a master clock signal (MCLK) generated by the 8X305 is used to latch either the I/O-enabling address or the I/O data into peripheral devices connected to the $\overline{\text{IV}}$ bus. MCLK can also be used to synchronize any external logic with timing circuits of the 8X305. To summarize the action, the first half of the instruction cycle deals primarily with input functions and the second half is mostly concerned with output functions.


INPUT PHASE
OUTPUT PHASE
1st QUARTER 50ns
2nd QUARTER 50ns
3rd QUARTER 50ns
4th QUARTER 50ns
INPUT INSTRUCTION, DECODE INSTRUCTION AND, IF REQUIRED, FETCH NEW DATA
LATCH AND PROCESS INPUT DATA
NEXT INSTRUCTION ADDRESS, GENERATE CONTROL SIGNALS, AND SETUP I/O DATA FOR OUTPUT
LATCH I/O ENABLING ADDRESS OR I/O DATA INTO SELECTED PERIPHERAL.
MCLK (ACTIVE STATE)
WF15230S

NOTES:

1. New instruction must be accepted and latched at end of first quarter cycle.
2. The I/O data latches are open for the first two quarter cycles, that is, for 100ns.
3. The address changes during third quarter cycle.
4. $\overline{IV}$ bus drivers are active (turned on) during third and fourth quarter cycles.

Figure 4. Instruction Cycle and MCLK with: Crystal = 10MHz and Cycle Time = 200ns.

Table 3.

↓ MSB LSB ↓

BIT POSITIONS →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	OPCODE			OPERAND(S)												

Table 4.

- **Data Control —**
  - **ADD**
  - **AND**
  - **XOR**
  - } **Arithmetic and Logic Operations**
  - **MOVE**
  - **XMIT**
  - } **Movement of Data and Constants**
- **Program Control —**
  - **XEC**
  - **NZT**
  - **JMP**
  - } **Branch or Test**

INSTRUCTION SET

General Format and Operating Principles

The 16-bit instruction word (I_0 through I_{15}) from program storage is input to the instruction register (Figure 1) and is subsequently decoded to implement the events to occur during the current instruction cycle.

The general format for each instruction word is shown in Table 3.

The 3-bit operation code (OPCODE) define any one of eight classes of instructions; variations within each class are specified by the remaining thirteen operand bits. The eight instruction classes can be separated into two control areas — *data* and *program*; general functions within these areas are as shown in Table 4.

Instruction Fields

As shown in Table 5, each instruction word consists of an operation code (OPCODE) field and from one to three operand fields. The possible operand fields are: Source (S), Destination (D), Rotate/Length (R/L), Literal (J), and Address (A). The OPCODE and operand fields are described in the paragraphs that follow the table.

Table 5. Functional Description of Instruction Set

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE — SEE FIGURE 4		
		CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
CLASS = **MOVE** OPCODE = **0** OPERATION = **(S)** → **D**				
Register-to-Register 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S R D S = $00_8 - 17_8$ D = $00_8 - 07_8$, $11_8 - 17_8$	Move content of internal register specified by S-field to internal register specified by D-field. Prior to the "MOVE" operation, right-rotate contents of internal source register by octal value (0 through 7) defined by the R-field.	SC	L	H if D = 07_8, 17_8
		WC	L	L
		$\overline{\text{LB}}$	H	L if D = 07_8
		$\overline{\text{RB}}$	H	L if D = 17_8
Register-to-$\overline{\text{IV}}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S L D D_1 ⋮ D_0 S = $00_8 - 17_8$ D = $20_8 - 37_8$	Move contents of internal register specified by the S-field to the $\overline{\text{IV}}$ bus. Before outputting on $\overline{\text{IV}}$ bus, data is shifted as specified by the least significant octal digit of the D-field and the bits specified by the L-field are merged with the latched I/O data.	SC	L	L
		WC	L	H
		$\overline{\text{LB}}$	L if D = $20_8 - 27_8$	L if D = $20_8 - 27_8$
		$\overline{\text{RB}}$	L if D = $30_8 - 37_8$	L if D = $30_8 - 37_8$
$\overline{\text{IV}}$ Bus-to-Register (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S L D S_1 ⋮ S_0 S = $20_8 - 37_8$ D = $00_8 - 07_8$, $11_8 - 17_8$	Move right-rotated $\overline{\text{IV}}$ bus (source) data specified by the S-field to internal register specified by the D-field. The L-field specifies the length of source data starting from the LSB-position and, if less than 8 bits, the remaining bits are filled with zeros.	SC	L	H if D = 07_8, 17_8
		WC	L	L
		$\overline{\text{LB}}$	L if S = $20_8 - 27_8$	L if D = 07_8
		$\overline{\text{RB}}$	L if S = $30_8 - 37_8$	L if D = 17_8
$\overline{\text{IV}}$ Bus-to-$\overline{\text{IV}}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S L D S_1 ⋮ S_0 D_1 ⋮ D_0 S = $20_8 - 37_8$ D = $20_8 - 37_8$	Move right-rotated $\overline{\text{IV}}$ bus (source) data specified by the S-field to the I/O latches. Before outputting on $\overline{\text{IV}}$ bus, shift data as specified by the D-field; then merge source and latched I/O data as specified by the L (length) field.	SC	L	L
		WC	L	H
		$\overline{\text{LB}}$	L if S = $20_8 - 27_8$	L if D = $20_8 - 27_8$
		$\overline{\text{RB}}$	L if S = $30_8 - 37_8$	L if D = $30_8 - 37_8$
CLASS = **ADD** OPCODE = **1** OPERATION = **(S) + (AUX)** → **D**				
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are ADDed to the source data. If there is a "carry" from MSB, then R10 (OVF) = 1 (overflow), otherwise OVF = 0.	Same as MOVE instruction class		
CLASS = **AND** OPCODE = **2** OPERATION = **(S) ∧ (AUX)** → **D**				
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are ANDed with source data.	Same as MOVE instruction class		
CLASS = **XOR** OPCODE = **3** OPERATION = **(S) ⊕ (AUX)** → **D**				
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are Exclusively ORed with source data.	Same as MOVE instruction class		
CLASS = **XEC** OPCODE = **4** OPERATION = **Refer to Description**				
Register Immediate 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S J S = $00_8 - 17_8$ J = $000_8 - 377_8$	Execute instruction at current page address offset by J (literal) + (S). Return to normal instruction flow unless a branch is encountered. Execute instruction at an address determined by replacing the low-order 8 bits of the Address Register with the following derived sum: Value of literal (J-field) plus contents of internal register specified by S-field The PC is not incremented and the overflow status (OVF) is not changed.	SC	L	L
		WC	L	L
		$\overline{\text{LB}}$	H	H
		$\overline{\text{RB}}$	H	H
$\overline{\text{IV}}$ Bus Immediate (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S L J S_1 ⋮ S_0 S = $20_8 - 37_8$ J = $00_8 - 37_8$	Execute instruction at an address determined by replacing the low-order 5 bits of Address Register with the following derived sum: 5-bit value of literal (J-field) plus value of rotated source data specified by S-field. The L-field specifies the length of source data starting from the LSB position and, if less than 8 bits, the remaining bits are filled with zeros; the Program Counter is not incremented and the overflow status (OVF) is not changed.	SC	L	L
		WC	L	L
		$\overline{\text{LB}}$	L if S = $20_8 - 27_8$	H
		$\overline{\text{RB}}$	L if S = $30_8 - 37_8$	H

Table 5. Functional Description of Instruction Set (Continued)

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE — SEE FIGURE 4		
		CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
CLASS = **NZT** OPCODE = **5** OPERATION = **Refer to Description**				
Register Immediate 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S J $S = 00_8 - 17_8$ $J = 000_8 - 377_8$	If data specified by the S-field is not equal to zero, jump to current page address offset by value of J-field; otherwise, increment the Program Counter. If contents of internal register specified by S-field is non-zero, transfer to address determined by replacing the low-order 8 bits of Address Register and Program Counter with "J", otherwise, increment PC.	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{RB}$	H	H
$\overline{IV}$ Bus Immediate (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE S L J S_1 ⋮ S_0 $S = 20_8 - 37_8$ $J = 00_8 - 37_8$	If right-rotated and masked IV bus is non-zero, transfer to address determined by replacing low-order 5 bits of Address Register and Program Counter with "J", otherwise, increment PC. (The L-field specifies the length of source I/O data starting from the LSB-position and, if less than 8 bits, the remaining bits are filled with zeros.)	SC	L	L
		WC	L	L
		$\overline{LB}$	L if $S = 20_8 - 27_8$	H
		$\overline{RB}$	L if $S = 30_8 - 37_8$	H
CLASS = **XMIT** OPCODE = **6** OPERATION = **J → D**				
XMIT, Register 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE D J $D = 00_8 - 06_8, 11_8, 14_8 - 16_8$ $J = 000_8 - 377_8$	Store 8-bit value specified by "J" into register specified by "D".	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{RB}$	H	H
XMIT, $\overline{IV}$ Bus Address 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE D J $D = 07_8, 17_8$ $J = 000_8 - 377_8$	Enable I/O device on the bank specified by "D", whose address is the 8-bit integer specified by "J". Address "J" is stored in register "D".	SC	L	H
		WC	L	L
		$\overline{LB}$	H	L if $D = 07_8$
		$\overline{RB}$	H	L if $D = 17_8$
XMIT 8 Bits Immediate, $\overline{IV}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE D J $D = 12_8 - 13_8$ $J = 000_8 - 377_8$	Store value of 8-bit integer in the previously enabled I/O port, at the bank destination ($\overline{LB}$ or $\overline{RB}$) specified by "D". Contents of R12 or R13 remain unchanged.	SC	L	L
		WC	L	H
		$\overline{LB}$	H	L if $D = 12_8$
		$\overline{RB}$	H	L if $D = 13_8$
XMIT Variable Bit Field Immediate, $\overline{IV}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE D L J D_1 ⋮ D_0 $D = 20_8 - 37_8$ $J = 00_8 - 37_8$	Transmit Least Significant "L" bits of "J" field to "L-bit" field of $\overline{IV}$ bus specified by "D"; if "L" is greater than 5 bits, the MSB bits of destination field is filled with zeros.	SC	L	L
		WC	L	H
		$\overline{LB}$	L if $D = 20_8 - 27_8$	L if $D = 20_8 - 27_8$
		$\overline{RB}$	L if $D = 30_8 - 37_8$	L if $D = 30_8 - 37_8$
CLASS = **JMP** OPCODE = **7** OPERATION = **Refer to Description**				
Address Immediate 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE A $A = 00000_8 - 17777_8$	Jump to address in program storage specified by A-field; this address is loaded into the Address Register and the Program Counter.	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{RB}$	H	H

NOTE:

S_0 specifies the LSB of rotated input data field

S_1 specifies the bank of $\overline{IV}$ bus from which source data will be input

D_0 specifies bit position in I/O device with which LSB of processed data will be aligned, and

D_1 specifies the bank of $\overline{IV}$ bus which will be the destination.

Table 6. Octal Addresses and Source/Destination Fields for 8X305 Registers

ADDRESS	REGISTER DESIGNATION	SOURCE	DESTINATION	ADDRESS	REGISTER DESIGNATION	SOURCE	DESTINATION
00_8	R0 (AUX) — General purpose register	X	X	10_8	R10 (OVF — Overflow register)	X	
01_8	R1 — General purpose register	X	X	11_8	R11 — General purpose register	X	X
02_8	R2 — General purpose register	X	X	12_8	R12 — General purpose register (Note)	X	X
03_8	R3 — General purpose register	X	X	13_8	R13 — General purpose register (Note)	X	X
04_8	R4 — General purpose register	X	X	14_8	R14 — General purpose register	X	X
05_8	R5 — General purpose register	X	X	15_8	R15 — General purpose register	X	X
06_8	R6 — General purpose register	X	X	16_8	R16 — General purpose register	X	X
07_8	R7 — Special purpose register (refer to next paragraph)	X	X	17_8	R17 — Special purpose register (refer to next paragraph)	X	X

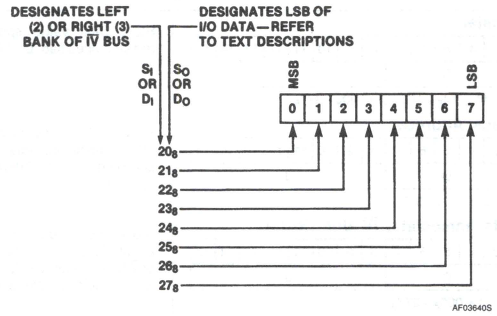
NOTE:
R12 and R13 function as general purpose working registers for all operations except transmit (XMIT). During a transmit instruction where R12 or R13 is the destination, the 8-bit "J" field is immediately transferred to the $\overline{IV}$ bus; for this operation, the contents of the designated register remain unchanged.

Operations Code Field. The 3-bit OPCODE field specifies one of eight classes of 8X305 instructions; octal designations for this field and operands for each instruction class are shown in Table 5.

Source (S) and Destination (D) Fields. The 5-bit "S" and "D" fields specify the source and destination, respectively, for whatever operation is defined by the OPeration CODE. The "S" and/or "D" fields can specify an internal 8X305 register or any one-to-eight bit field within an I/O device; octal values and source/destination field assignments for all internal registers are shown in Table 6.

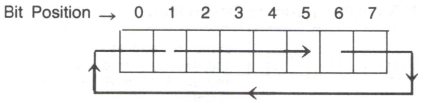
In instructions where $R7_8$ (IVL) or $R17_8$ (IVR) is specified as the destination, the 8-bit value is output on the $\overline{IV}$ bus as an I/O device address or memory location; register R7 selects the Left Bank and register R17 selects the Right Bank. The results are also stored into the specified internal register ($R7_8$ or $R17_8$) and may later be accessed as source data. When the $\overline{IV}$ bus is specified as a source and/or destination, the "S" and "D" fields are split into two parts, that is,

- Source (S) = S_1, S_0 and Destination (D) = D_1, D_0 where,
  S_0 specifies the LSB of rotated input data field
  S_1 specifies the bank of $\overline{IV}$ bus from which source data will be input
  D_0 specifies bit position in I/O device with which LSB of processed data will be aligned and
  D_1 specifies the bank of $\overline{IV}$ bus which will be the destination.


DESIGNATES LEFT (2) OR RIGHT (3) BANK OF $\overline{IV}$ BUS
DESIGNATES LSB OF I/O DATA—REFER TO TEXT DESCRIPTIONS
S_1 OR D_1
S_0 OR D_0
MSB
LSB
0
1
2
3
4
5
6
7
20_8
21_8
22_8
23_8
24_8
25_8
26_8
27_8
AF03640S

NOTES:
1. The field length of 0-to-8 bits is specified by the "L" field.
2. For the Right Bank, $30_8 - 37_8$ perform equivalent I/O functions.

RIGHT-ROTATE FUNCTION



Rotate (R) and Length (L) Field. The 3-bit R/L field performs one of two functions, specifying either the field length (L) for I/O operations or a right-rotate (R) for internal operations. For a given instruction, the specified function depends upon the contents of the Source (S) and Destination (D) fields.

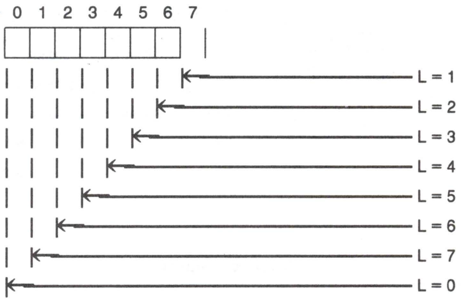
When an internal register is specified by both the source and destination fields, the "R" field is invoked and it specifies a right-rotate of the data specified in the "S" field (see accompanying diagram.) The source-register data (up to 8 bits) is right-rotated during the "input phase" of the instruction cycle (Figure 4). This function is always performed prior to any ALU operation. (*Note: The right-rotate function is implemented on the bus and not in the source register.*)

When either or both of the source and destination fields specify a variable-length I/O data field, the "L" field specifies the length of the I/O data field (see following diagram). If the source field specifies an $\overline{\text{IV}}$ address ($20_8 - 37_8$) and the destination field specifies an internal register ($00_8 - 07_8$, $11_8 - 17_8$), the "L" field specifies the length of source data; the source data is formed by right-rotating the $\overline{\text{IV}}$ bus data according to the source address and then masking result as specified by the "L" field. If length is less than 8 bits, all remaining bits are set to zero prior to processing data in the ALU. If the source field specifies an internal register ($00_8 - 17_8$) and the destination field specifies $\overline{\text{IV}}$ bus data ($20_8 - 37_8$), the "L" field specifies the length of the destination data. To form the destination data, the ALU output is left-shifted according to the destination address and then masked to the required length (see $\overline{\text{IV}}$ DATA LENGTH SPECIFICATION). The destination data is merged with data in the I/O latches to finalize the $\overline{\text{IV}}$ bus data. Hence, a one-to-eight bit destination data field can be inserted into the existing 8-bit I/O port without modifying surrounding bits. If both the source and destination fields specify $\overline{\text{IV}}$ bus data ($20_8 - 37_8$), the "L" field specifies the length of both the source and destination data.

To form the source data, the $\overline{\text{IV}}$ bus input data is right-rotated according to the source address and then masked to the required length — see $\overline{\text{IV}}$ DATA LENGTH SPECIFICATION. If length is less than 8 bits, all remaining bits are set to zero before processing in the ALU. To form the destination data, the ALU output is left-shifted according to the destination address and masked to the required length specification. The destination data is then merged into the $\overline{\text{IV}}$ bus data that was used to obtain the source; thus, if the source and destination addresses are on the same bank, the $\overline{\text{IV}}$ bus data written to the destination I/O Port appears unmodified, except for bits changed during the shift-and-mask operations. If the source and destination addresses refer to different banks, the destination I/O Port is changed to contain the contents of the source I/O Port in those bit positions not affected by the destination data.

J Field. The 5-bit or 8-bit "J" field is used to load a literal value (contained in the instruction) into a register, into a variable I/O data field, or to modify the low-order bits of the Program Counter. The bit length of the "J" field is implied by the "S" and "L" fields in the XEC, NZT, and XMIT instructions, based on the following conditions:

- When the Source (S) field specifies an internal register, the literal value of the "J" field is an 8-bit binary number.
- When the Source (S) field specifies a variable I/O data field, the literal value of the "J" field is a 5-bit binary number.



$\overline{\text{IV}}$ DATA LENGTH SPECIFICATION
(No Rotate Function Specified)

A Field. The 13-bit "A" field is an address field which allows the 8X305 to directly branch to any of the 8192 locations in Program Storage memory.

Formation of Instruction Address

The Address Register and Program Counter are used to generate addresses for accessing an instruction from program storage. The instruction address is formed in one of the following ways:

- For all except the JMP, XEC, and a "satisfied" NZT instruction, the Program Counter is incremented by one and placed in the Address Register.
- For the JMP instruction, the 13-bit "A" field contained in the JMP instruction word replaces the contents of both the Address Register and the Program Counter.
- For the XEC instruction, the Address Register is loaded with bits from the Program Counter modified as follows:
  XEC using $\overline{\text{IV}}$ Bus Data — low-order 5 bits of ALU output replaces counterpart bits in Address Register.

  XEC using Data from Internal Register — low-order 8 bits of ALU output replaces counterpart bits in Address Register.

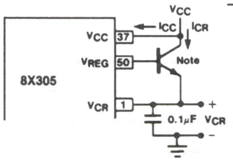
The Program Counter is not modified for either of the above conditions.

- For a "satisfied" NZT instruction, the low-order 5 bits (NZT source is $\overline{\text{IV}}$ bus data) or low-order 8 bits (NZT source is an internal register) of both the Address Register and Program Counter are loaded with the literal value specified by the "J" field of instruction word.

Data Addressing

The source and/or destination addresses of the data to be operated upon are specified as part of the instruction word. As shown earlier, source/destination addresses are specified using a 5-bit code ($00_8 - 37_8$). When the most significant octal digit is a "0" or "1", the source and/or destination address is an internal register; if the most significant digit is a 2 or 3, an $\overline{\text{IV}}$ bus operation is indicated — 2 specifying a Left-Bank ($\overline{\text{LB}}$) operation and 3 specifying a Right-Bank ($\overline{\text{RB}}$) operation. The least significant octal digit (0 through 7) indicates either a specific internal register address or positioning information for the least significant bit when specifying $\overline{\text{IV}}$ bus data. Referring to Table 5, AUXiliary register R0 (00_8) is the implied source of the second argument for the ADD, AND, and XOR operations. IVL register R7 and IVR register R17 (destination addresses 07_8 and 17_8, respectively) provide a means of routing enabling address information to I/O peripherals. With IVL or IVR specified as the destination address, data is placed on the $\overline{\text{IV}}$ bus during the output phase of the instruction cycle; simultaneously, a Select Command (SC) is generated to inform all I/O devices that information on the $\overline{\text{IV}}$ bus is to be considered as an I/O address. Since the contents of IVL and IVR are preserved, either register may later be accessed as a source of data.

Control outputs $\overline{\text{LB}}$ and $\overline{\text{RB}}$ are used to partition I/O bus devices into two fields of 256 addresses. With $\overline{\text{LB}}$ in the active-low state and a source address of $20_8 - 27_8$, the left bank of I/O devices are enabled during the input phase of the instruction cycle. With $\overline{\text{RB}}$ in the active-low state and a source address of $30_8 - 37_8$, the right bank of devices are enabled. During the output phase, $\overline{\text{LB}}$ is low if the destination address is 07_8 or $20_8 - 27_8$, whereas $\overline{\text{RB}}$ is low if the destination address is 17_8 or $30_8 - 37_8$. Each address field ($\overline{\text{LB}}$ and $\overline{\text{RB}}$) can have a different I/O device selected, that is, data can be transferred from a device in one bank to a device in the other in one instruction cycle.



PARAMETER	CONDITIONS	LIMITS
h_{fe}	$V_{CE} = 2V$; $100mA < I_C < 500mA$	> 50
VBE_{ON}	$V_{CE} = 5V$; $I_C = 500mA$	< 1V
V_{CESAT}	$I_C = 500mA$; $I_B = 50mA$	< 0.5V
BV_{CEO}		> 15V
f_t		> 30MHz

LD05661S

NOTE:
Typical approved parts — 2N5320, 2N5337

Voltage Regulator

DESIGN PARAMETERS

Hardware design of an 8X305-based system largely consists of the following operations:

- Selecting and interfacing a Program Storage device — ROM, PROM, etc.
- Selecting and interfacing input/output devices — RAM, Ports, and other 8-bit addressable I/O devices.
- Choosing and implementing System Clock — Capacitor-Controlled, Crystal-Controlled, or Externally-Driven.
- Selection of an off-chip series-pass transistor.

VOLTAGE REGULATOR

All internal logic of the 8X305 is powered by an on-chip voltage regulator that requires an external series-pass transistor. Electrical specifications for the off-chip power transistor and a typical hook-up are shown in the accompanying diagram. To minimize lead inductance, the transistor should be as close as possible to the 8X305 package and the emitter should be AC-grounded via a 0.1μf ceramic capacitor.

All information required for easy implementation of these design requirements is provided under the following captions:

- Ordering Information
- Voltage Regulator
- DC Characteristics
- AC Characteristics
- Timing Considerations
- Clock Considerations
- $\overline{\text{HALT}}/\overline{\text{RESET}}$ Logic

ABSOLUTE MAXIMUM RATINGS Storage Temperature (T_{STG}) rating are from −65°C to +150°C

SYMBOL	PIN	DESCRIPTION	RATING	UNIT
V_{CC}	V_{CC}	Supply voltage	+7.0	V
	X1, X2	Crystal input voltage	2.0	V
	All other pins	Logic input voltage	5.5	V

DC ELECTRICAL CHARACTERISTICS (Commercial Part) $4.75V \leqslant V_{CC} \leqslant 5.25V$, $0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT	COMMENTS
			Min	Typ	Max		
V_{CC}	Supply voltage		4.75	5	5.25	V	
V_{IH}	High-level input voltage		0.9 2		2 5.5	V	X1 and X2 All other pins
V_{IL}	Low-level input voltage				0.5 0.8	V	X1 and X2 All other pins
V_{OH}	High-level output voltage	V_{CC} = min; I_{OH} = −3mA	2.4			V	
V_{OL}	Low-level output voltage	V_{CC} = min; I_{OL} = 6mA V_{CC} = min; I_{OL} = 16mA			0.55 0.55	V	A_0 through A_{12} All other outputs
V_{CR}	Regulator voltage	V_{CC} = 5V		3.1 2.9		V	T_A = 0°C T_A = 70°C
V_{IC}	Input clamp voltage	V_{CC} = min; I_{IN} = −10mA			−1.5	V	Crystal inputs X1 and X2 do not have internal clamp diodes
I_{IH}	High-level input current	V_{CC} = max V_{IH} = 0.9V V_{IH} = 4.5V			4 50	mA μA	X1 and X2 All other pins
I_{IL}	Low-level input current	V_{CC} = max; V_{IL} = 0.4V			−3 −0.2 −1.6 −0.4	mA	X1 and X2 $\overline{IV0}$ − $\overline{IV7}$ I0 − I15 $\overline{HALT}$ and $\overline{RESET}$
I_{OS}	Short circuit output current	V_{CC} = max; (Note: At any time, no more than one output should be connected to ground.)	−30		−140	mA	All output pins
I_{CC}	Supply current	V_{CC} = max			180 195	mA	T_A = 70°C T_A = 0°C
I_{REG}	Regulator control	V_{CC} = 5.0V	−10		−25	mA	Max available base drive for series-pass transistor
I_{CR}	Regulator current	V_{CC} = max			200 230	mA	T_A = 70°C T_A = 0°C

NOTES:

1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. All voltages measured with respect to ground terminal.

AC ELECTRICAL CHARACTERISTICS

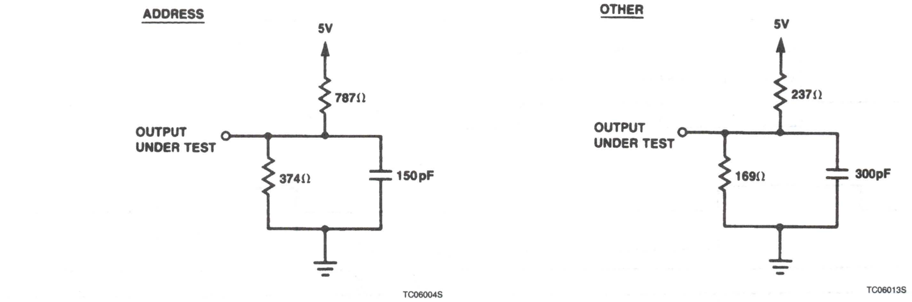
(Commercial Part) Conditions: $4.75V \leqslant V_{CC} \leqslant 5.25V$; $0°C \leqslant T_A \leqslant 70°C$
Loading: (See test circuits)

SYMBOL	PARAMETER (NOTE 1)	LIMITS (INSTRUCTION CYCLE TIME = 200ns)			LIMITS (INSTRUCTION CYCLE TIME > 200ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
t_{PC}	Processor cycle time	200			200			ns	
t_{CP}	X1 clock period	100			100			ns	
t_{CH}	X1 clock high time	50			50			ns	
t_{CL}	X1 clock low time	50			50			ns	
t_{MCL}	MCLK low delay	15		40	15		40	ns	
t_W	MCLK pulse width	35		55	$T_{4Q} - 15$		$T_{4Q} + 5$	ns	Note 2
t_{DD}	Input data to output data	70		105	70		105	ns	
t_{MHS}	MCLK falling edge to $\overline{HALT}$ falling edge			30			$T_{1Q} - 20$	ns	Note 2
t_{MHH}	$\overline{HALT}$ hold time (MCLK falling edge)	65			$T_{1Q} + 15$			ns	Note 2
t_{ACC}	Program storage access time			60				ns	
t_{IO}	I/O port output enable time ($\overline{LR}/\overline{RB}$ to valide $\overline{IV}$ data input)			30				ns	
t_{MAS}	MCLK falling edge to address stable			140			$T_{1Q} + T_{2Q} + 40$	ns	Notes 2, 3, & 4
t_{IA}	Instruction to address			140			$T_{2Q} + 90$	ns	Notes 2, 3 & 5
t_{IVA}	Input data to address			85			85	ns	Notes 3 & 6
t_{MIS}	MCLK falling edge to instruction stable			25			$T_{1Q} - 25$	ns	Notes 2 & 7
t_{MIH}	Instruction hold time (MCLK falling edge)	55			$T_{1Q} + 5$			ns	Notes 2 & 8
t_{MWH}	MCLK falling edge to SC/WC rising edge	105		125	$T_{1Q} + T_{2Q} + 5$		$T_{1Q} + T_{2Q} + 25$	ns	Note 2
t_{MWL}	MCLK falling edge to SC/WC falling edge	2		15	2		15	ns	
t_{MIBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	5		25	5		25	ns	
t_{IIBS}	Instruction to $\overline{LB}/\overline{RB}$ (Input phase)			25			25	ns	
t_{MOBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	115		145	$T_{1Q} + T_{2Q} + 15$		$T_{1Q} + T_{2Q} + 45$	ns	Note 2
t_{MIDS}	MCLK falling edge to input data stable			55			$T_{1Q} + T_{2Q} - 45$	ns	Note 2
t_{MIDH}	Input data hold time (MCLK falling edge)	115			$T_{1Q} + T_{2Q} + 15$			ns	Note 2
t_{MODH}	Output data hold time (MCLK falling edge)	11			11			ns	
t_{MODS}	Output data stable (MCLK falling edge)	123		150	$T_{1Q} + T_{2Q} + 23$		$T_{1Q} + T_{2Q} + 50$	ns	Note 2
t_{ODSM}	Output data stable (MCLK rising edge)	10			$t_{3Q} - 40$			ns	Note 2

NOTES:

1. X1 and X2 inputs are driven by an external pulse generator with an amplitude of 1.5 volts; all timing parameters are measured at this voltage level.
2. Respectively, T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q} represent time intervals for the first, second, third, and fourth quarter cycles.
3. Capacitive loading for the address bus is 150 picofarads.
4. T_{MAS} is obtained by forcing a valid instruction and an I/O bus input to occur earlier than the specified minimum set up time.
5. T_{IA} is obtained by forcing a valid instruction input to occur earlier than the minimum set up time.
6. T_{IVA} is obtained by forcing a valid I/O bus input to meet the minimum set up time.
7. T_{MIS} represents the setup time required by internal latches of the 8X305. In system applications, the instruction input may have to be valid before the worst-case set up time in order for the system to respond with a valid I/O bus input that meets the I/O bus input set up time (T_{IDS} and T_{MIDS}).
8. T_{MIH} represents the hold time required by internal latches of the 8X305. To generate proper $\overline{LB}/\overline{RB}$ signals, the instruction must be held valid until the address bus changes.

AC TEST CIRCUITS


ADDRESS
5V
787Ω
OUTPUT UNDER TEST
374Ω
150pF
TC06004S
OTHER
5V
237Ω
OUTPUT UNDER TEST
169Ω
300pF
TC06013S

NOTE:
Load capacitance includes Test Jig and Probe Capacitance

TIMING CONSIDERATIONS (Commercial Part)

As shown in the AC CHARACTERISTICS table for the commercial part, the minimum instruction cycle time is 200ns; whereas, the maximum is determined by the on-chip oscillator frequency and can be any value the user chooses. With an instruction cycle time of 200ns, the part can be characterized in terms of absolute values; these are shown in the first "LIMITS" column of the table. When the instruction cycle time is greater than 200ns, certain parameters are cycle-time dependent; thus, these parameters are specified in terms of the four quarter cycles (T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q}) that make up one instruction cycle — see 8X305 TIMING DIAGRAM. As the time interval for each instruction cycle increases (becomes greater than 200ns), the delay for all parameters that are cycle-time dependent is likewise increased. In some cases, these delays have a significant impact on timing relationships and other areas of systems design; subsequent paragraphs describe these timing parameters and reliable methods of calculation.

Timing parameters for the 8X305 are normally measured with reference to MCLK.

System determinants for the instruction cycle time are:

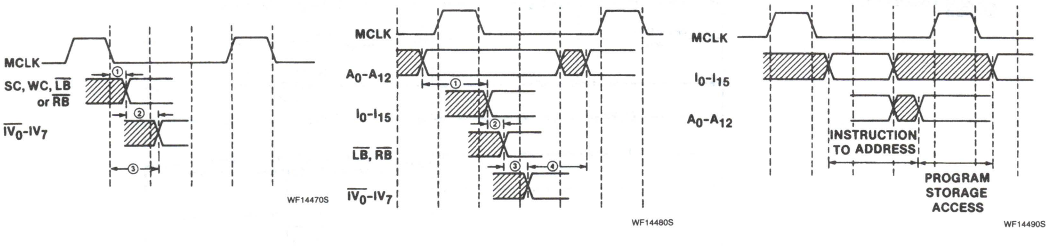
- Propagation delays within the 8X305
- Access time of Program Storage
- Enable time of the I/O port

Normally, the instruction cycle time is constrained by one or more of the following conditions:

Condition 1 — Instruction or MCLK to $\overline{LB}/\overline{RB}$ (input phase) plus I/O port access time (TIO) $\leqslant$ $\overline{IV}$ data setup time (Figure 5a).

Condition 2 — Program storage access time (TACC) plus instruction to $\overline{LB}/\overline{RB}$ (input phase) plus I/O port access time (TIO) plus $\overline{IV}$ data (input phase) to address $\leqslant$ instruction cycle time (Figure 5b).

Condition 3 — Program storage access time plus instruction to address $\leqslant$ instruction cycle time (Figure 5c).


MCLK
SC, WC, $\overline{LB}$ or $\overline{RB}$
$\overline{IV}_0$-$\overline{IV}_7$
①
②
③
WF14470S
MCLK
A_0-A_{12}
I_0-I_{15}
$\overline{LB}$, $\overline{RB}$
$\overline{IV}_0$-$\overline{IV}_7$
①
②
③
④
WF14480S
MCLK
I_0-I_{15}
A_0-A_{12}
INSTRUCTION TO ADDRESS
PROGRAM STORAGE ACCESS
WF14490S

NOTES:
① MCLK to $\overline{LB}/\overline{RB}$ (input phase) or instruction to $\overline{LB}/\overline{RB}$ (input phase).
② I/O port access (TIO).
③ $\overline{IV}$ data setup time (referenced to MCLK).

a. Condition #1

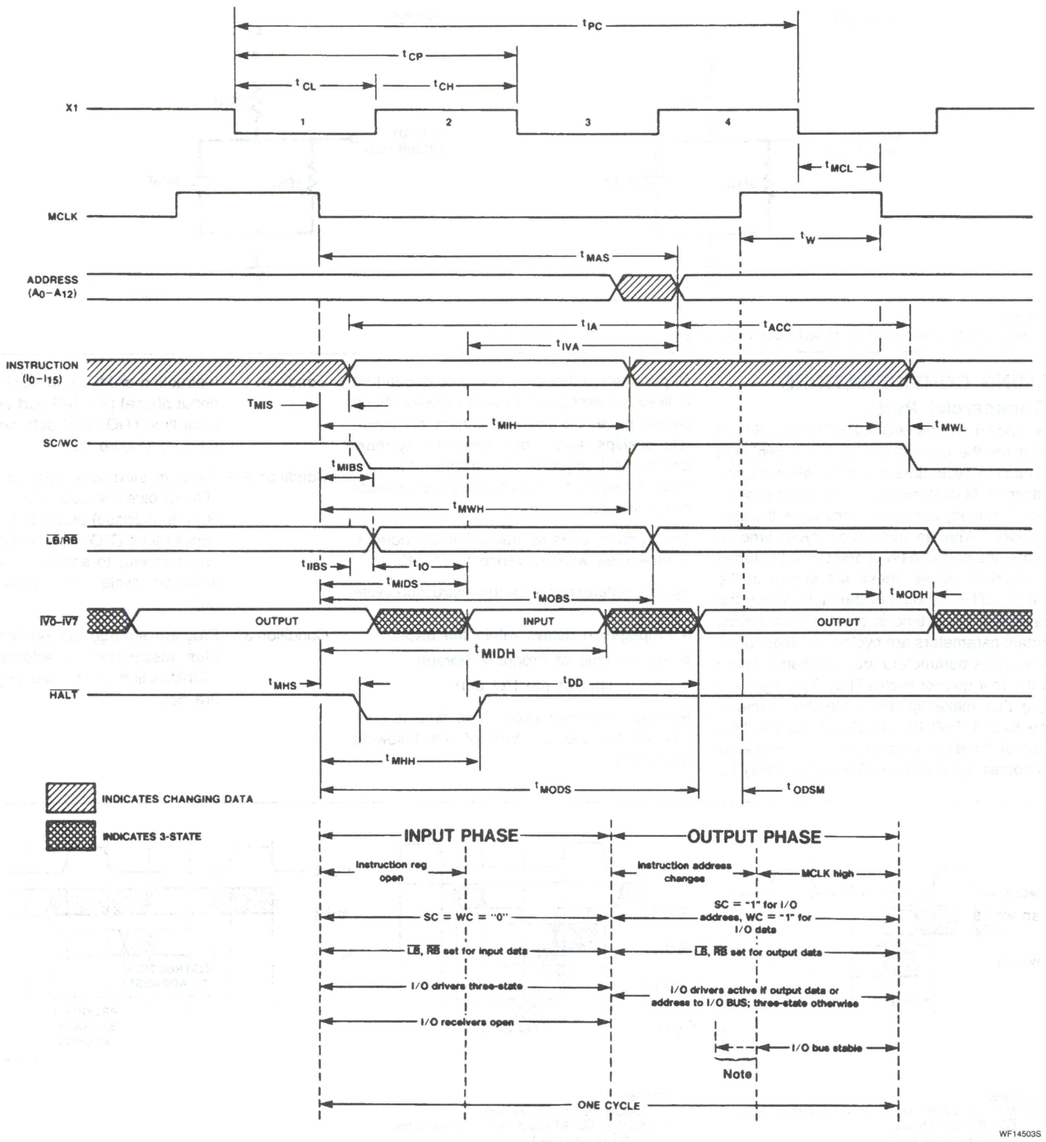
NOTES:
① Program storage access time.
② MCLK to $\overline{LB}/\overline{RB}$ (input phase) or instruction to $\overline{LB}/\overline{RB}$ (input phase).
③ I/O port access (TIO).
④ $\overline{IV}$ data (input phase to address.)

b. Condition #2

c. Condition #3

Figure 5. Constraints of 8X305 Instruction Cycle Time

8X305 TIMING DIAGRAM



tPC
tCP
tCL
tCH
X1
1
2
3
4
tMCL
MCLK
tW
tMAS
ADDRESS (A0–A12)
tIA
tACC
tIVA
INSTRUCTION (I0–I15)
TMIS
tMIH
tMWL
SC/WC
tMIBS
tMWH
LB/RB
tIIBS
tIO
tMIDS
tMOBS
tMODH
IV0–IV7
OUTPUT
INPUT
OUTPUT
tMIDH
tMHS
tDD
HALT
tMHH
tMODS
tODSM
INDICATES CHANGING DATA
INDICATES 3-STATE
INPUT PHASE
OUTPUT PHASE
Instruction reg open
Instruction address changes
MCLK high
SC = WC = "0"
SC = "1" for I/O address, WC = "1" for I/O data
LB, RB set for input data
LB, RB set for output data
I/O drivers three-state
I/O drivers active if output data or address to I/O BUS; three-state otherwise
I/O receivers open
I/O bus stable
Note
ONE CYCLE
WF14503S

NOTE:
For an instruction cycle time greater than 200ns, the I/O bus can be stable sometime within the third quarter (T_{3Q}) cycle.

From condition #1 and with an instruction cycle time of 200ns, the I/O port access time (TIO) can be calculated as follows:

TMIBS + TIO $\leqslant$ TMIDS
transposing, TIO $\leqslant$ TMIDS = TMIBS
substituting, TIO $\leqslant$ 55ns − 25ns
result, TIO $\leqslant$ 30ns

Using 30ns for TIO, the constraint imposed by condition #1 can also be used to calculate the minimum cycle time:

TMIBS + TIO $\leqslant$ TMIDS
thus, 25ns + 30ns $\leqslant T_{1Q} + T_{2Q} - 45$
25ns + 30ns $\leqslant$ 1/2 cycle − 45

therefore, the worst-case instruction cycle time is 200ns. With subject parameters referenced to X1, the same calculations are valid:

TIBS + TIO + TIDS $\leqslant$ 1/2 cycle
thus, 45ns + 30ns + 25ns $\leqslant$ 1/2 cycle

therefore, the worst-case instruction cycle time is again 200ns. From condition #2 and with an instruction cycle time of 200ns, the program storage access time can be calculated:

TACC + TIIBS + TIO + TIVA $\leqslant$ 200ns
transposing, TACC $\leqslant$ 200ns − TIIBS − TIO − TIVA
substituting, TACC $\leqslant$ 200ns − 25ns − 30ns − 85ns
thus, TACC $\leqslant$ 60ns

hence, for instruction cycle time of 200ns, a program storage access time of 60ns is implied. The constraint imposed by condition #3 can be used to verify the maximum program storage access time:

TIA + TACC $\leqslant$ Instruction Cycle
thus, TACC $\leqslant$ 200ns − 140ns
and, TACC $\leqslant$ 60ns

confirming that a program storage access time of 60ns is satisfactory.

For an instruction cycle time of 200ns and a program storage access time of 60ns (Condition #2/Figure 5b), the instruction should be valid at the falling edge of MCLK. This relationship can be derived by the following equation:

200ns − TMAS − TACC
= 200ns − 140ns − 60ns
= 0ns

It is important to note that, during the input phase, the beginning of a valid $\overline{LB}/\overline{RB}$ signal is determined by either the instruction to $\overline{LB}/\overline{RB}$ delay (TIIBS) or the delay from the falling edge of MCLK to $\overline{LB}/\overline{RB}$ (TMIBS). Assuming the instruction is valid at the falling edge of MCLK and adding the instruction-to-$\overline{LB}/\overline{RB}$ delay (TIIBS = 25ns), the $\overline{LB}/\overline{RB}$ signal will be valid 25ns after the falling edge of MCLK. With a fast program storage memory and with a valid instruction before the falling edge of MCLK — the $\overline{LB}/\overline{RB}$ signal will, due to the TMIBS delay, still be valid 25ns after the falling edge of MCLK. Using a worst-case instruction cycle time of 200ns, the user cannot gain a speed advantage by selecting a memory with faster access time. Under the same conditions, a speed advantage cannot be obtained by using an I/O port with fast access time (TIO) because the address bus will be stable 55ns (TAS) after the beginning of the third quarter cycle — no matter how early the $\overline{IV}$ data input is valid.

CLOCK CONSIDERATIONS

The on-chip oscillator and timing-generation circuits of the 8X305 can be controlled by any one of the following methods:

Capacitor — if timing is not critical
Crystal — if precise timing is required
External Drive — if application requires that the 8X305 be driven from a system clock

Capacitor Timing. A non-polarized ceramic or mica capacitor with a working voltage equal to or greater than 25V is recommended. The lead lengths of capacitor should be approximately the same and as short as possible; also, the timing circuits should not be in close proximity to external sources of noise. For various capacitor (C_X) values, the cycle time can be approximated as:

C_X (in pF)	APPROXIMATE CYCLE TIME
100	300ns
200	500ns
500	1.1μs
1000	2.0μs

Crystal Timing. When a crystal is used, the on-chip oscillator operates at the resonant frequency (f_o) of the crystal. The series-resonant quartz crystal connects to the 8X305 via pins 10 (X1) and 11 (X2). The lead lengths of the crystal should be approximately equal and as short as possible. Also, the timing circuits should not be in close proximity to external sources of noise. The crystal should be hermetically sealed (HC type can) and have the following electrical characteristics:

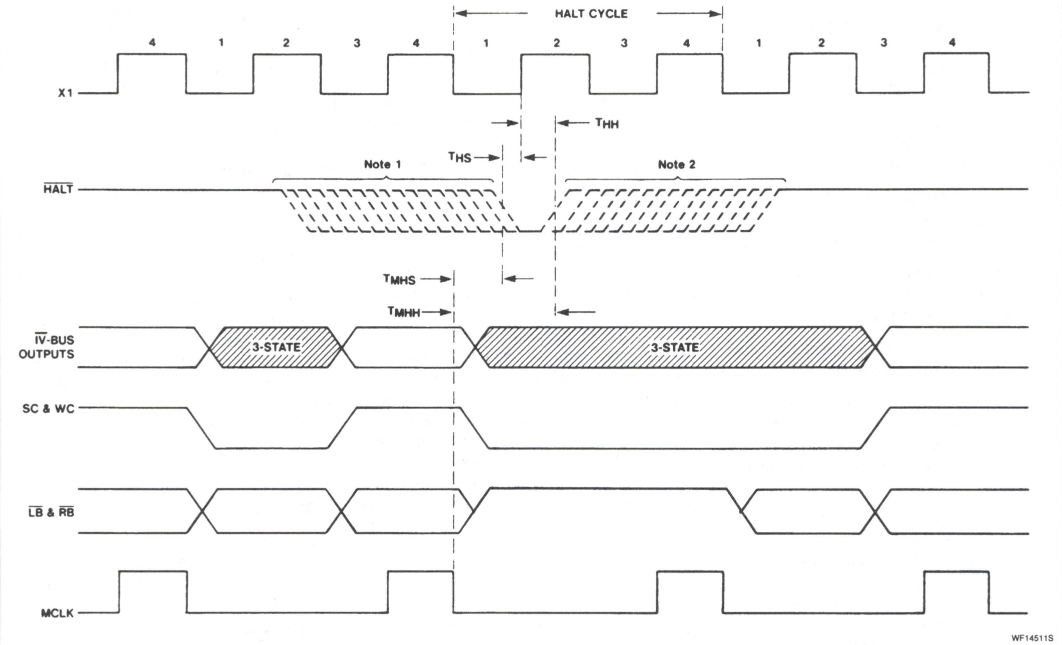
Type — Fundamental mode, series resonant
Impedance at Fundamental — 35Ω max.
Impedance at Harmonics and Spurs — 50Ω min.

The resonant frequency (f_o) of the crystal is related to the desired cycle time (T) by the equation: $f_o = 2/T$; thus, for a cycle time of 200ns, f_o = 10MHz.

$\overline{HALT}$ Logic

The $\overline{HALT}$ signal is sampled via internal chip logic at the end of the first internal quarter of each instruction cycle. If, when sampled, the $\overline{HALT}$ signal is active-low, a halt is immediately executed and the current instruction cycle is terminated. However, the halt cycle does not inhibit MCLK nor does it affect any internal registers of the 8X305. As long as the $\overline{HALT}$ line is active-low, the SC and WC lines are low (inactive), the Left Bank ($\overline{LB}$)/Right Bank ($\overline{RB}$) signals are high (inactive), and the $\overline{IV}$ bus remains in the 3-State mode of operation. Normal operation resumes at the next cycle in which $\overline{HALT}$ is high when sampled (see $\overline{HALT}$ TIMING DIAGRAM.)

HALT TIMING DIAGRAM


HALT CYCLE
4
1
2
3
4
1
2
3
4
1
2
3
4
X1
THH
THS
Note 1
Note 2
HALT
TMHS
TMHH
IV-BUS OUTPUTS
3-STATE
3-STATE
SC & WC
LB & RB
MCLK
WF14511S

NOTES:
1. The $\overline{\text{HALT}}$ signal can switch from High to Low at any time during this interval.
2. The $\overline{\text{HALT}}$ signal can switch from Low to High at any time during this interval.

TIMING DESCRIPTIONS:

T_{HS} — setup time from $\overline{\text{HALT}}$ to X1 (independent of instruction cycle time)
T_{HH} — hold time from X1 to $\overline{\text{HALT}}$ (independent of instruction cycle time)
T_{MHS} — setup time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)
T_{MHH} — hold time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)

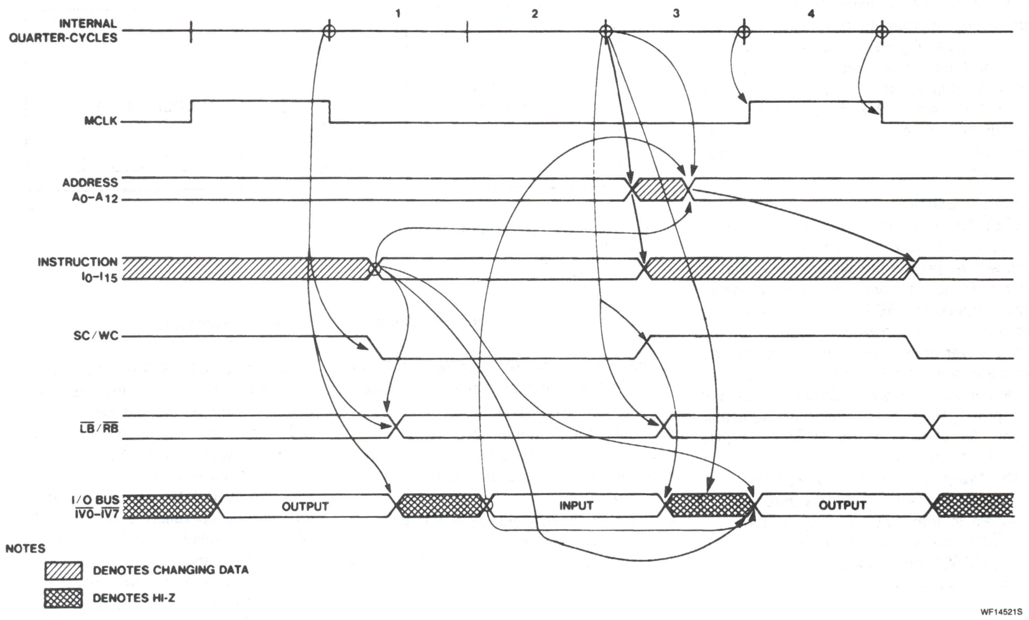

INTERNAL QUARTER-CYCLES
1
2
3
4
MCLK
ADDRESS A0–A12
INSTRUCTION I0–I15
SC/WC
LB/RB
I/O BUS IV0–IV7
OUTPUT
INPUT
OUTPUT
NOTES
DENOTES CHANGING DATA
DENOTES HI-Z
WF14521S

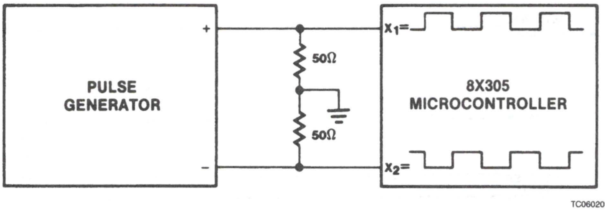
Figure 7. Timing Relationships of 8X305 I/O Signals

Using an External Clock. The 8X305 can be synchronized with an external clock by simply connecting appropriate drive circuits to the X1/X2 inputs. Figure 8 shows how the on-chip oscillator can be driven from the complementary outputs of a pulse generator. In applications where the Microcontroller must be driven from a master clock, the X1/X2 lines can be interfaced to TTL logic as shown in Figure 9.

$\overline{\text{RESET}}$ Logic

$\overline{\text{RESET}}$ (pin 43) can be driven from a high (inactive) state to a low (active) state at any time with respect to the system clock, that is, the reset function is asynchronous. To ensure proper operation, $\overline{\text{RESET}}$ must be held low (active) for one full instruction time. When the line is driven from a high state to an active-low state, several events occur — the precise instant of occurrence is basically a function of the propagation delay for that particular event. As shown in the $\overline{\text{RESET}}$ TIMING DIAGRAM, these events are:

- The Program Counter and Address Register are set to address zero and remain in that state as long as the $\overline{\text{RESET}}$ line is low. Other than PC and AR, $\overline{\text{RESET}}$ does not affect other internal registers.



PULSE GENERATOR CHARACTERISTICS:

$Z_{OUT} = 50\Omega$
RISETIME $\leqslant$ 10ns
$V_{OUT} = 0$ to 1.5V
Skew $\leqslant$ 10ns

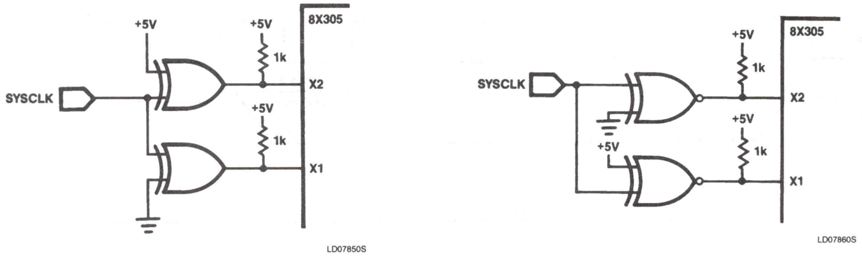
Figure 8. Clocking With a Pulse Generator

- The input/output ($\overline{\text{IV}}$) bus goes 3-State and remains in that condition as long as the $\overline{\text{RESET}}$ line is low.
- The Select Command and Write Command signals are driven low and remain low as long as the $\overline{\text{RESET}}$ line is low.
- The Left Bank/Right Bank ($\overline{\text{LB}}/\overline{\text{RB}}$) signals are forced high asynchronously for the period in which the $\overline{\text{RESET}}$ line is low.

During the time $\overline{\text{RESET}}$ is active-low, MCLK is inhibited. Moreover, if the $\overline{\text{RESET}}$ line is driven low during the last two quarter cycles, MCLK may be shortened for that particular machine cycle. When $\overline{\text{RESET}}$ line is driven high (inactive) — one quarter to one full instruction cycle later, MCLK appears just before normal operation is resumed. The $\overline{\text{RESET}}$/MCLK relationship is clearly shown by "B" in the timing diagram. As long as the $\overline{\text{RESET}}$ line is active-low, the $\overline{\text{HALT}}$ signal (described next) is not sampled by internal logic of the 8X305.

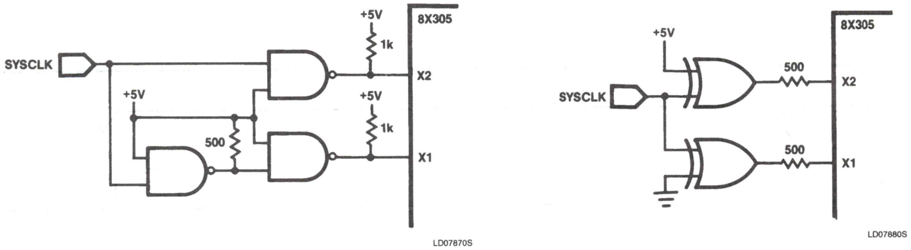
1. Using a 74LS136 Quad Execlusive-OR gate (open collector).

2. Using a 74LS266 Quad Exclusive-NOR gate (open collector).


+5V
+5V
8X305
1k
SYSCLK
X2
+5V
1k
X1
LD07850S
+5V
8X305
SYSCLK
1k
X2
+5V
+5V
1k
X1
LD07860S

3. Using a 74F38* Quad NAND gate (open collector).

4. Using a 74LS286 Quad Exclusive-OR gate.


+5V
8X305
1k
SYSCLK
X2
+5V
+5V
1k
500
X1
LD07870S
+5V
8X305
500
X2
SYSCLK
500
X1
LD07880S

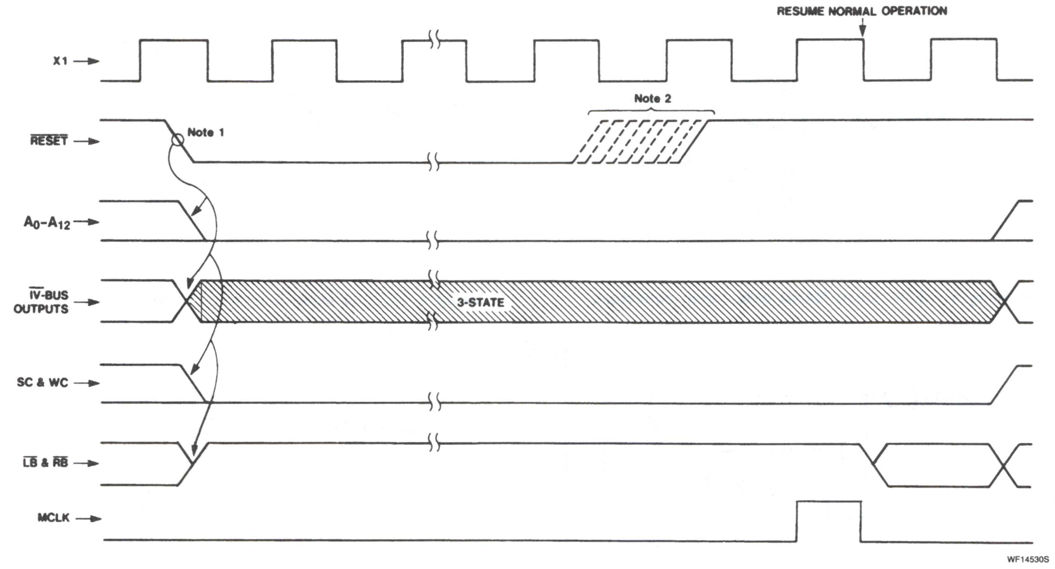
TTL DRIVER CHARACTERISTICS:
Fall Time ≤ 10ns
Skew Between Complementary Outputs ≤ 10ns

NOTES:

1. All circuits, as drawn, preserve the phase relationship of SYSCLK to X1.
2. The resistor values of 1k ohms for open-collector pull-ups and 500 ohms for active pull-up series resistors are calculated to be optimum for all opening conditions and silicon varations.
3. The 8X305 clock may be driven by circuits other than those shown here. The circuits shown however, have been tested under conditions in excess of those that will be found in a normal system. Exclusive-OR/NOR type gates were selected to minimize the skew between the X1 and X2 inputs.
4. 74LS38 and 74S38 gates were tested in addition to the 74F38 chip. These were found to be the least robust configurations of all those tested, although they did work over normal operating conditions abd beyond. When failure occured it was due to excessive skew between X1 and X2 caused by the inverting gate in the X1 leg.

Figure 9. Clocking With TTL

RESET TIMING DIAGRAM


RESUME NORMAL OPERATION
X1
Note 2
RESET
Note 1
A0-A12
IV-BUS OUTPUTS
3-STATE
SC & WC
LB & RB
MCLK
WF14530S

NOTES:

1. A High to Low transition of the RESET signal will force the Address Bus to an all-zero configuration.
2. The RESET signal can switch from Low to High at any point within this time interval and, in all cases, MCLK will occur at least one-quarter cycle time later as shown.

Signetics

8X310
Interrupt Control Coprocessor

Product Specification

Microprocessor Products

FEATURES

- **Three prioritized interrupts**
- **Subroutine handling capabilities**
- **4-level LIFO stack for return address storage**
- **Interrupt masking by software and hardware**
- **Stack full flag**
- **Directly compatible with 8X305 Microcontroller**
- **Bipolar ISL (Integrated Schottky Logic) and low-power Schottky technology**
- **Single +5V power supply**
- **0.6″, 40-pin DIP**

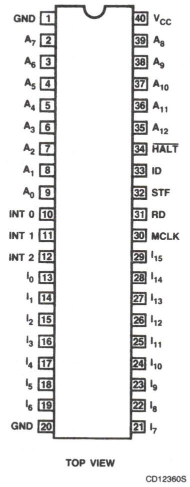
DESCRIPTION

The Signetics 8X310 Interrupt Control Coprocessor (ICC) supports the 8X305 Microcontroller in systems that are interrupt driven and those that require subroutine handling capabilities.

As shown in Figure 1, the ICC provides three prioritized interrupt request lines: INT 0 (highest priority), INT 1 and INT 2. A low-to-high transition applied to any of these input lines latches in an interrupt request which may be serviced when sampled by the ICC once each instruction cycle of the Microcontroller. When an interrupt request is serviced, the ICC forces the Microcontroller to jump to one of three fixed locations in program memory; instruction addresses 4, 5, and 6 correspond to INT 0, INT 1 and INT 2. At each of these addresses, the user programs a JMP instruction to another address where the user's interrupt service routine begins.

PIN CONFIGURATIONS

N, I PACKAGES


GND 1
A_7 2
A_6 3
A_5 4
A_4 5
A_3 6
A_2 7
A_1 8
A_0 9
INT 0 10
INT 1 11
INT 2 12
I_0 13
I_1 14
I_2 15
I_3 16
I_4 17
I_5 18
I_6 19
GND 20
40 V_{CC}
39 A_8
38 A_9
37 A_{10}
36 A_{11}
35 A_{12}
34 $\overline{HALT}$
33 ID
32 STF
31 RD
30 MCLK
29 I_{15}
28 I_{14}
27 I_{13}
26 I_{12}
25 I_{11}
24 I_{10}
23 I_9
22 I_8
21 I_7
TOP VIEW
CD12360S

During interrupt servicing, the ICC also stores the proper return address into a four deep, Last-In-First-Out (LIFO) stack. At the conclusion of the interrupt service routine, the user program instructs the ICC to return to the main program at the location previously stored in the stack. The return operation is implemented by coding a special RETURN instruction which is decoded directly off the instruction bus by the ICC. There are five such special instructions relating to interrupt and subroutine handling functions performed by the ICC. These instruction codes are all treated as non-operational instructions (NOPs) by the Microcontroller.

An internal one-bit mask is used to inhibit interrupt servicing. Whenever the mask is set, the ICC does not respond to any pending interrupt requests; however, any requests remain latched for future servicing. The mask can be set and cleared either by the user program or automatically during certain ICC functions. The special instructions SET MASK and CLEAR MASK are provided for user control. The Interrupt Disable input also inhibits interrupt request servicing.

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
40-Pin plastic DIP	N8X310N
40-Pin ceramic DIP	N8X310I

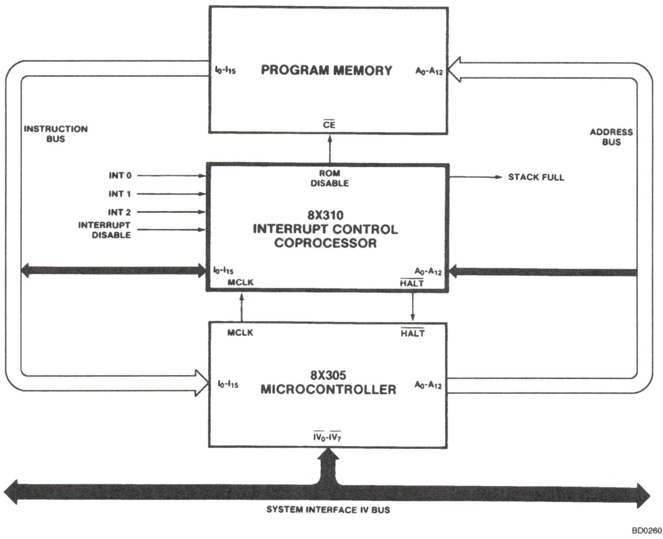

PROGRAM MEMORY
I_0-I_{15}
A_0-A_{12}
$\overline{CE}$
INSTRUCTION BUS
ADDRESS BUS
ROM DISABLE
STACK FULL
INT 0
INT 1
INT 2
INTERRUPT DISABLE
8X310 INTERRUPT CONTROL COPROCESSOR
I_0-I_{15}
MCLK
A_0-A_{12}
$\overline{HALT}$
MCLK
$\overline{HALT}$
8X305 MICROCONTROLLER
I_0-I_{15}
A_0-A_{12}
$\overline{IV_0}$-$\overline{IV_7}$
SYSTEM INTERFACE IV BUS
BD02601S

Figure 1. Typical System Connections Using ICC

Interrupt Control Coprocessor 8X310

PIN DESCRIPTION

PIN NO.	IDENTIFIER	FUNCTION
1, 20	GND	Ground. (Note: The printed circuit board should not use the ICC as a bridge for external ground.)
2 – 9, 35 – 39	$A_7 - A_0$, $A_{12} - A_8$	Program address input lines from Microcontroller. Active high. A_0 is MSB.
10 – 12	INT 0 – INT 2	Interrupt request input pins. INT 0 has the highest priority and INT 2 the lowest — Edge-triggered on a low-to-high transition.
13 – 19, 21 – 29	$I_0 - I_6$, $I_7 - I_{15}$	Bidirectional instruction bus; I_0 is MSB. When acting as an input, the ICC decodes the instruction flow (binary pattern on $I_0 - I_{15}$) between program storage and the Microcontroller. During an interrupt or return cycle, the ICC outputs a JMP instruction to the Microcontroller via these lines — refer to FUNCTIONAL OPERATION of ICC.
30	MCLK	Master CLocK — active high input from 8X305 Microcontroller used for a timing reference and system synchronization.
31	RD	ROM (or PROM) Disable — active high output used to disable normal program storage so that the ICC can force an instruction to the Microcontroller.
32	STF	STack Full — active high output. When the LIFO stack is full, STF goes high and remains high until at least one register in the 4-level stack is empty.
33	ID	Interrupt Disable — active high. When this input pin is driven high, servicing of all interrupt requests is suspended.
34	$\overline{\text{HALT}}$	Active low output. Suspends all processing operations of the Microcontroller during period when the source of instruction data is changing between the ICC and program storage.
40	V_{CC}	+5V power supply.

Interrupt Control Coprocessor 8X310

FUNCTIONAL OPERATION

Basic Functions

The ICC performs the three general functions indicated below.

Function 1: Provides a means for the 8X305 Microcontroller to respond to interrupt requests by diverting the program flow of the 8X305 Microcontroller to the proper interrupt service routine or, in the case of a subroutine, the ICC stores the return address in the 4-level LIFO stack (Figure 2).

Function 2: Returns the user to the proper point in the main program for both interrupt and subroutine activities.

Function 3: Provides both automatic and programmed masking capabilities.

Interrupt Requests and Priority Considerations

An interrupt is requested when any one of the ICC input pins INT 0, INT 1, or INT 2 undergoes a low-to-high transition; this request is temporarily stored in an internal edge-triggered latch that corresponds to the affected interrupt input. The interrupt request latches are part of the Priority and Mask Logic shown in Figure 2. Unless masked or otherwise disabled, the ICC samples these latches once each instruction cycle. Any or all of the latches may be set when sampled by the ICC; however, only the interrupt of highest priority will be serviced — the remaining interrupts will be held in queue. Thus, if INT 0, INT 1 and INT 2 simultaneously compete for service, INT 0 is the first to be serviced followed, in order, by INT 1 and INT 2; likewise, if INT 1 and INT 2 compete for service, INT 1, being of higher priority will be serviced first. The CLEAR INTERRUPT instruction resets all interrupt request latches without affecting an interrupt service routine that is already in progress.

The highest priority interrupt request will be serviced when sampled by the ICC provided interrupts in general are not inhibited and a previous interrupt of equal or higher priority is not currently being serviced. The general masking of interrupts is discussed later. To determine priorities, the ICC keeps track of any interrupt that is serviced until the corresponding service routine returns. A subsequent interrupt request may interrupt a service routine in progress only if it is of a higher priority than that of the current interrupt being serviced. If, for example, INT 1 is requested and serviced, then before its service routine finishes, a request on INT 0 can be serviced as a second level interruption. However, a request on INT 2 or a second request on INT 1 must wait until the original INT 1 service routine returns. The interrupt service routine that was interrupted will resume execution at the point of interruption when the higher priority service routine returns (i.e. in the same manner as when returning to the main program).

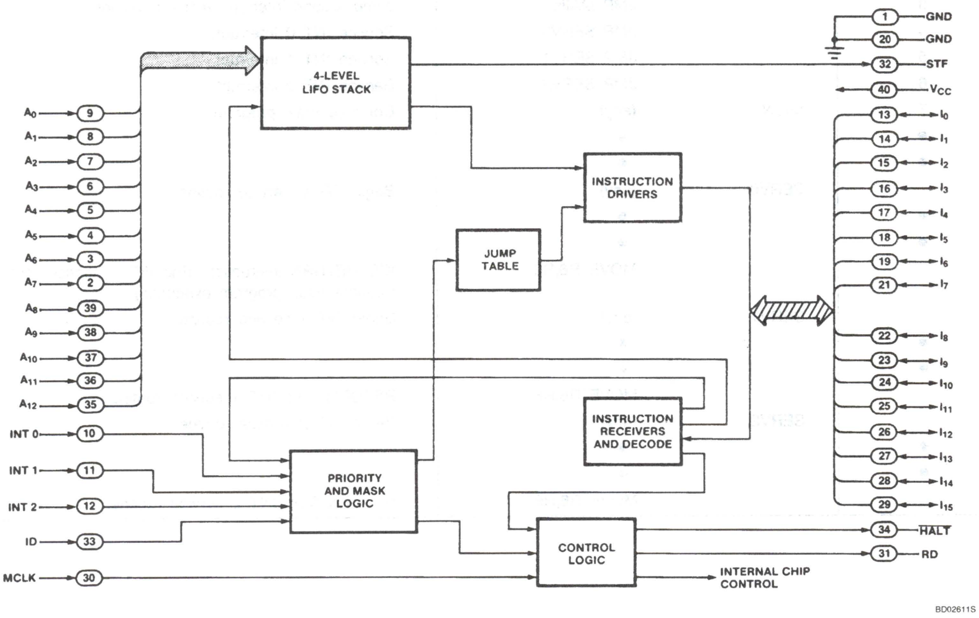

4-LEVEL LIFO STACK
INSTRUCTION DRIVERS
JUMP TABLE
INSTRUCTION RECEIVERS AND DECODE
PRIORITY AND MASK LOGIC
CONTROL LOGIC
INTERNAL CHIP CONTROL
A_0 9
A_1 8
A_2 7
A_3 6
A_4 5
A_5 4
A_6 3
A_7 2
A_8 39
A_9 38
A_{10} 37
A_{11} 36
A_{12} 35
INT 0 10
INT 1 11
INT 2 12
ID 33
MCLK 30
1 GND
20 GND
32 STF
40 V_{CC}
13 I_0
14 I_1
15 I_2
16 I_3
17 I_4
18 I_5
19 I_6
21 I_7
22 I_8
23 I_9
24 I_{10}
25 I_{11}
26 I_{12}
27 I_{13}
28 I_{14}
29 I_{15}
34 $\overline{HALT}$
31 RD
BD02611S

Figure 2. 8X310 Interrupt/Control Coprocessor — Functional Block Diagram

Interrupt Servicing

Interrupts are sampled only at the conclusion of an instruction cycle while the next instruction is being fetched from Program Memory.

When an interrupt request is serviced, the following general steps are performed:

- Address of the instruction that would normally be executed next is pushed into the 4-level LIFO Stack (Figure 2) for subsequent return to the main program.
- The ICC disables program storage and forces a JMP instruction onto the Instruction bus of the 8X305 Microcontroller. (Note: Because of timing considerations, the HALT signal is driven low to suspend operation of the Microcontroller for one instruction cycle; this permits the source of instruction data to change from program storage to the ICC without conflict.) The JMP instruction from the ICC transfers the Microcontroller to one of the three fixed program locations shown below. In each of these addresses, the user will normally store a JMP instruction to the interrupt service routine for that particular interrupt. Details of these operations are described later.

INT 0 ---------------- Address 4

INT 1 ---------------- Address 5

INT 2 ---------------- Address 6

Return from Interrupt Service Routine

Upon completion of the interrupt service routine, the user codes the special RETURN instruction. When executed, the ICC performs the following steps

- The return address is popped from the LIFO stack.
- The ICC disables program storage and forces a JMP instruction onto the Microcontroller instruction bus with the return address from the stack. (The HALT signal is driven low for one instruction cycle.)
- The JMP instruction from the ICC transfers the Microcontroller to the instruction that was about to execute at the time the interrupt was taken

A typical structure for a user program which handles interrupts is shown in the following example:

ADDRESS	INSTRUCTION		COMMENT
0		(any)	First instruction executed after system reset
•		•	
•		•	
3		JMP MAIN	Jump around interrupt vector locations.
4		JMP SERV0	Service INT 0 interrupt.
5		JMP SERV1	Service INT 1 interrupt.
6		JMP SERV2	Service INT 2 interrupt.
7	MAIN	(any)	Continue main program.
•		•	
•		•	
	SERV0	(any)	Begin INT 0 service routine.
•		•	
•		•	
		MOVE R6,R6	ICC RETURN instruction. End INT 0 service routine (resume main program execution).
	SERV1	(any)	Begin INT 1 service routine.
•		•	
•		•	
		MOVE R6,R6	RETURN from INT 1 service routine.
	SERV2	(any)	Begin INT 2 service routine.
•		•	
•		•	
		MOVE R6,R6	RETURN from INT 2 service routine.

Table 1. Instruction Set for ICC

INSTRUCTION	INSTRUCTION CODES		8X305 OPERATION	DESCRIPTION OF ICC OPERATION
	Assembler	Binary		
		I_0 = MSB I_{15} = LSB ↓ ↓		
SET MASK	MOVE R5,R5	000 00101 000 00101	NOP	When executed, sets interrupt mask, thus inhibiting all interrupt servicing.
CLEAR MASK	MOVE R4,R4	000 00100 000 00100	NOP	When executed, clears interrupt mask for all interrupts.
RETURN	MOVE R6,R6	000 00110 000 00110	NOP	When executed returns program to address at top of LIFO stack.
PUSH	MOVE R3,R3	000 00011 000 00011	NOP	Pushes "address + 2" onto stack if PUSH is programmed on odd address in program memory and "address + 1" if PUSH is programmed on even address.
CLEAR INTERRUPT	MOVE R2,R2	000 00010 000 00010	NOP	Clears all interrupt requests; an interrupt service routine that is in progress is unaffected.

Instruction Set

The five instructions shown in Table 1 allow the user to efficiently manage both the interrupt and subroutine capabilities of the ICC in an 8X310/8X305-based system. When an ICC instruction appears in the program, it is interpreted as a NOP by the 8X305 but is captured and decoded by the ICC to perform the desired function. The capture-and-decode functions of the chip are automatic. Assembly and object codes for each ICC instruction are shown in Table 1.

Interrupt Masking Operations

Certain operations performed by the ICC and also some system considerations require that program execution not be interrupted for a specified interval of time. The servicing of interrupts by the ICC can be inhibited in a number of ways. Any time interrupts are inhibited, the ICC ignores any latched interrupt requests. However, the interrupt request latches are not cleared so that any previously pending requests remain latched. Also, during an interval when interrupt servicing is inhibited, any new interrupt signals received will get latched. As soon as interrupt servicing is enabled, any latched requests can be serviced on a priority basis.

The primary means of inhibiting interrupt servicing is the internal one-bit mask (latch). This mask can be set (to inhibit interrupts) or cleared under control of the user program using the special ICC instructions SET MASK and CLEAR MASK — See Table 1. With these instructions, segments of the user program can be isolated so as to proceed without interruptions. Frequently, uninterruptable segments are needed at the very beginning of the user program (initialization routine) and at the beginning of, or throughout an interrupt service routine. To facilitate this, the ICC automatically sets the mask whenever the Microcontroller executes address zero (typically resulting from a system reset) and whenever the ICC services an interrupt. The ICC also automatically clears the mask after performing a RETURN operation from an interrupt service routine; a RETURN from a subroutine does not affect the status of the interrupt mask.

The Interrupt Disable (ID) input pin may also be used to inhibit interrupt servicing. Interrupt servicing remains disabled as long as a high level is applied to the input. The ID input has no effect, however, on the status of the internal interrupt mask.

To ensure proper program flow, the ICC suspends interrupt servicing momentarily during certain situations. During the cycle in which the Microcontroller encounters an XEC (Execute) instruction an interrupt will not be serviced. This is because the XEC causes the Microcontroller to issue an address of an instruction to be executed out of the sequence of normal program flow. This would not be a valid address.

Interrupts are also suspended during execution of a PUSH or RETURN instruction and the instruction immediately following. This ensures proper operation of the LIFO stack. In addition, no interrupts are latched or serviced and no special ICC instructions are decoded at address zero which resets the ICC.

Subroutine Calling

The ICC provides for subroutine calling by storing the proper return address into the LIFO stack under control of the user program. Two instructions are required to implement a subroutine call — a PUSH instruction executed by the ICC and a JMP to the subroutine executed by the Microcontroller. The PUSH instruction is normally programmed at an odd-numbered address in program memory immediately followed by the JMP. When the PUSH instruction is executed, the ICC finds the address of the next instruction (JMP to subroutine) on the Microcontroller's address bus, internally changes the least-significant bit to one (effectively adds one to the address) and stores this into the stack. Program execution proceeds normally and the Microcontroller makes the jump to the beginning of the subroutine. The subroutine may be located at any convenient place in program memory.

Upon completion of the subroutine, the user codes the RETURN instruction in the same manner as for an interrupt service routine. At that point, the ICC forces the Microcontroller to resume execution of the main program at the instruction immediately following the JMP-to-subroutine instruction.

The code for a typical subroutine call-and-return is shown in the following example:

ADDRESS	INSTRUCTION	COMMENT
X (any odd-numbered address)	MOVE R3,R3	PUSH instruction initiates subroutine call by causing ICC to push the address X + 2 onto the stack. (The PUSH instruction is interpreted as a NOP by the Microcontroller.)
X + 1 (even)	JMP SUBR	The Microcontroller JMPs to the beginning of the subroutine.
X + 2 (odd)	(any instruction)	Main program execution resumes here after RETURN from subroutine.
● ● ●	● ● ●	
SUBR (any address) ● ● ●	(any instruction) ● ● ●	Execution of subroutine starts here.
(any address)	MOVE R6,R6	RETURN instruction causes ICC to transfer program back to X + 2.

Stack Operation

The LIFO stack holds up to four 13-bit program addresses which allows the ICC to return from a subroutine or interrupt service routine. When all four stack locations are filled, the STack Full (STF) output pin is driven high and remains high until a RETURN (or reset) operation occurs. If an additional interrupt is serviced or subroutine called while the stack is full, the stack will overflow and the oldest return address will be overwritten and lost. That is, the stack retains the four most recent entries. After an overflow, the status of the STF output is not valid (until a reset operation occurs).

To prevent an interrupt from overflowing the stack, the user can connect the STF output directly to the Interrupt Disable (ID) input of the ICC. Then, even if the internal mask and priorities permit interrupt servicing, the interrupt request must still wait for the most recent service routine or subroutine to return.

Because subroutine calling is controlled explicitly by the user software, the user can always ensure that subroutine nesting alone could not overflow the stack. However, care must be taken whenever calling a subroutine from within an interrupt service routine since the number of remaining stack locations may vary at the time the interrupt is taken. If, for example, three stack locations are already filled (STF is low) at the time an interrupt is serviced, then a subroutine call executed within the interrupt service routine would cause the stack to overflow and the earliest return address to be lost.

As mentioned earlier, whenever a RETURN operation is performed from an interrupt service routine, the internal interrupt mask is automatically cleared. A RETURN from a subroutine, however, does not affect the status of the mask. To accomplish this, a flag bit is added to each of the four stack locations which records whether each address pushed into the stack is caused by an interrupt or a subroutine call. This flag is read during a RETURN operation to determine whether or not the interrupt mask is cleared. This allows interrupt servicing and subroutine calls to be intermixed in any order.

Initialization

The ICC decodes address zero as a reset command to perform certain initialization functions. (Zero is the first address generated after the Microcontroller is reset.) Specifically, the Instruction-bus drivers are placed in a high-impedance state, HALT output is set high, RD output is set low, and all interrupt request latches are cleared. The interrupt mask is set so that any initialization routine by the user will not be interrupted until a CLEAR MASK instruction (MOVE R4,R4) is executed. The LIFO stack is reset to an empty state and the STF output is set low.

SYSTEM TIMING RELATIONSHIPS

Interrupt Servicing

Interrupt servicing begins at the end of a Microcontroller instruction cycle when the 8X305's MCLK signal goes from low-to-high. Starting from this point, processing of the interrupt proceeds as follows:

From the rising edge of MCLK 1:

- Interrupt mask is set to inhibit other interrupts.
- HALT output is driven low to stop internal operation of the 8X305 Microcontroller for one instruction cycle; MCLK is unaffected. ROM Disable (RD) output is driven high to disable program memory.

From the falling edge of MCLK 1:

- Takes address of next instruction from address bus and pushes it onto the top of stack to be used as the return address to the main program.

From the rising edge of MCLK 2:

- The ICC forces a JMP onto the instruction bus to one of three fixed vector addresses:

INT 0 - - - - - - - - - - - - - - - - Address 4
INT 1 - - - - - - - - - - - - - - - - Address 5
INT 2 - - - - - - - - - - - - - - - - Address 6

- Releases HALT (high) which allows the Microcontroller to complete the JMP to the above specified vector location in program memory.

From the rising edge of MCLK 3:

- Instruction-bus drivers of the ICC are disabled.
- ROM Disable (RD) pin is cleared (low) enabling the program memory which resumes control of the Instruction bus.

Return Operation

When the interrupt service routine or subroutine is completed, the RETURN instruction initiates the following sequence of events:

From the rising edge of MCLK 1:

- Interrupts are temporarily inhibited through third MCLK cycle.
- HALT output is driven low to stop Microcontroller for one instruction cycle.
- RD output is set high to disable program memory.

From the rising edge of MCLK 2:

- HALT output is driven high (cleared).
- A JMP instruction to address stored at top of LIFO stack is forced onto the Instruction bus by the ICC. (The stack is popped.)

From the rising edge of MCLK 3:

- Instruction-bus drivers of the ICC are disabled.
- RD is cleared enabling the program memory.
- If returning from an interrupt service routine (condition recorded in extra stack bit) the interrupt mask is cleared; otherwise the mask remains unaffected.

Once the preceding return actions are completed, the 8X305 Microcontroller will resume execution of the instruction at the return address.

APPLICATION HINTS

- When programming an interrupt service routine or subroutine, certain system operations typically need to be considered. In many interrupt-driven systems, a handshake signal is required to acknowledge the servicing of an interrupt request. The acknowledge signal may be transmitted by the interrupt service routine using a standard I/O port from the 8X300 Family.
- If the user wants to allow a higher priority interrupt request to interrupt a service routine, then the CLEAR MASK instruction should be programmed (perhaps after completing any critical operations).
- For both service routines and subroutines, the user may need to save the contents of some or all of the working registers of the Microcontroller so that operation of the main program is not upset. Registers may be written out to a working storage RAM such as 8X350 near the beginning of the routine, and restored from RAM just before returning to the main program.
- Certain subroutine calling techniques may be used to increase the efficiency of the user program. As shown in the following examples, a subroutine can automatically be repeated two, three or four times, if desired, without programming a loop.

SUBROUTINE AUTOMATICALLY EXECUTES TWICE			
Address	**Instruction**		
X(even)	SUBR2	MOVE R3,R3	Push X + 1 onto stack.
X + 1(odd)		(start of subroutine)	
•		•	
•		•	
		MOVE R6,R6	RETURN — First time jumps to X+1; second time jumps back to main program.
SUBROUTINE AUTOMATICALLY EXECUTES THREE TIMES			
X(odd)	SUBR3	MOVE R3,R3	Push X + 2 onto stack.
X + 1(even)		MOVE R3,R3	Push X + 2 onto stack.
X + 2(odd)		(start of subroutine)	
•		•	
•		•	
SUBROUTINE AUTOMATICALLY EXECUTES FOUR TIMES			
X(even)	SUBR4	MOVE R3,R3	Push X + 1 onto stack.
X + 1(odd)		MOVE R3,R3	Push X + 3 onto stack.
X + 2(even)		NOP	
X + 3(odd)		(start of subroutine)	
•		•	
•		•	

- In a manner similar to the Microcontroller multi-way branch technique, one of several subroutines can be selected according to an index value.

SUBROUTINE CALL SELECTED BY VALUE IN R1			
Address	**Instruction**		
X(odd)		MOVE R3,R3	Push X + 2 onto stack.
X + 1(even)		XEC TABLE(R1)	Execute JMP at TABLE + (R1)
X + 2(odd)		(any)	Subroutine returns here.
•		•	
•		•	
(any)	TABLE	JMP SUB0	Call SUB0 if R1 = 0.
		JMP SUB1	Call SUB1 if R1 = 1.
		•	
		•	

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Rating	Unit
V_{CC}	Power supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_O	Off-state output voltage	+5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V$ (±5%); $0°C \leq T_A \leq 70°C$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				0.8	V
V_{OH}	High-level output voltage	V_{CC} = Min.; $I_{OH} = -1$ mA	2.4			V
V_{OL}	Low-level output voltage	V_{CC} = Min.; $I_{OL} = 8$ ma			0.55	V
V_{CL}	Input clamp-diode voltage	V_{CC} = Min; $I_{CL} = -10$ mA			−1.5	V
I_{IH}	High-level input current	V_{CC} = Max; $V_{IH} = 2.7$ V			100	μA
I_{IL}	Low-level input current	V_{CC} = Max; $V_{IL} = 0.4$ V			−550	μA
I_{OS}	Short circuit output current	V_{CC} = Max; $V_O = 0$ V	−15		−80	mA
I_{CC}	Supply current	V_{CC} = Max; $I_0 - I_{15}$ = High-Z				mA
		$T_A = 0°C$[2]			200	
		$T_A = 70°C$			185	

AC ELECTRICAL CHARACTERISTICS

V_{CC} = 5.0V (±5%); 0°C $\leqslant T_A \leqslant$ 70°C
Loading: See Test Loading Circuit

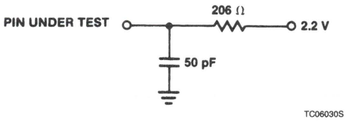
SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS			UNIT
		From	To		Min	Typ	Max	
Pulse Widths:								
t_{WIH}	Interrupt High	↑ INT_i	↓ INT_i		30			ns
t_{WIL}	Interrupt Low	↓ INT_i	↑ INT_i		35			ns
t_{WMH}	MCLK High	↑ MCLK	↓ MCLK	For all Functions	40			ns
Propagation Delays:								
t_{PRH}	RD High	↑ MCLK	↑ RD	Interrupt or Return			70	ns
t_{PRL}	RD Low	↑ MCLK	↓ RD	Interrupt or Return			15	ns
t_{PHL}	HALT Low	↑ MCLK	↓ HALT	Interrupt or Return			70	ns
t_{PHH}	HALT High	↑ MCLK	↑ HALT	Interrupt or Return			65	ns
t_{PSH}	Stack full High	↓ MCLK	↑ STF	Interrupt or Subroutine Call			105	ns
t_{PSL}	Stack full Low	↓ MCLK	↓ STF	Return or Reset			110	ns
Setup Times:								
t_{SIH}	Interrupt input setup[3]	↑ INT_i	↑ MCLK		35			ns
t_{SA}	Address setup	$A_0 - A_{13}$	↑ MCLK	Interrupt, Subroutine Call, or Reset	0			ns
t_{SC}	Instruction setup[5]	I_{0-15}	↓ MCLK	All Commands	Note 5			ns
t_{SD}	Interupt disable setup[3]	ID	↑ MCLK		30			ns
Hold and Reset Recovery Times:								
t_{HIL}	Interrupt low input hold[3]	↑ MCLK	↑ INT_i		15			ns
t_{HA}	Address hold	↓ MCLK	A_{0-13}	Subroutine Call or Reset	75			ns
t_{HC}	Instruction hold	↓ MCLK	I_{0-15}	All Commands	55			ns
t_{HD}	Interrupt disable hold[3]	↑ MCLK	ID		25			ns
t_{RI}	Interrupt reset recovery[4]	↓ MCLK	↑ INT_i	Reset or Cancel Command	70			ns
Output Enable/Disable Delays:								
t_{OEC}	Instruction output enable	↑ MCLK	I_{0-15}	Interrupt or Return			70	ns
t_{ODC}	Instruction output disable	↑ MCLK	I_{0-15}	Interrupt or Return			40	ns

NOTES:

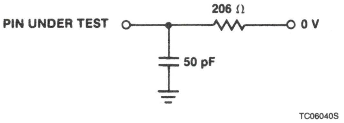
1. All electrical characteristics are guaranteed after and thermal equilibrium has been reached.
2. The 200mA value is worst case over the entire temperature range for the Commercial.
3. Parameters t_{SIH}, t_{HIL},T_{SD}, and t_{HD} are used only to determine whether an interrupt request will be serviced during the current or a subsequent instruction cycle. The INT_i and ID inputs are asynchronous and transitions on either input may safely occur at any time with respect to MCLK. A Low-to-High transition on INT_i occurring after t_{SIH} and before t_{HIL} means only that it cannot be determined for sure whether or not the interrupt request will be honored during the current instruction cycle. Similarly, transitions on ID between t_{SD} and t_{HD} make it uncertain as to whether or not masking applies during the current instruction cycle.
4. When clearing interrupt requests (including a reset operation), any new low-to-high transitions appearing at the INT_i inputs that occur before t_{RI} risk being cleared and therefore ignored; however, any transition after t_{RI} is certain to be latched.
5. t_{SC}(minimum) = 15 ns — t_{PRL}(actual).
   (The required instruction enable time for the program memory depends on the sum of the t_{PRL} and t_{SC}.)

TEST SETUPS

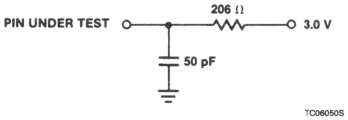
RD, STF, and $\overline{\text{HALT}}$ Outputs —


PIN UNDER TEST
206 Ω
2.2 V
50 pF
TC06030S

$I_0 - I_{15}$ Outputs (t_{OEC} and t_{ODC}) t_{PZH} and t_{PHZ}:


PIN UNDER TEST
206 Ω
0 V
50 pF
TC06040S

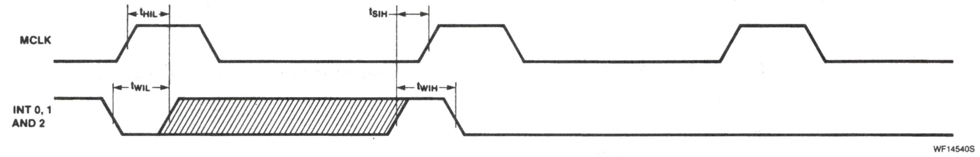
t_{PZL} and t_{PLZ}:


PIN UNDER TEST
206 Ω
3.0 V
50 pF
TC06050S

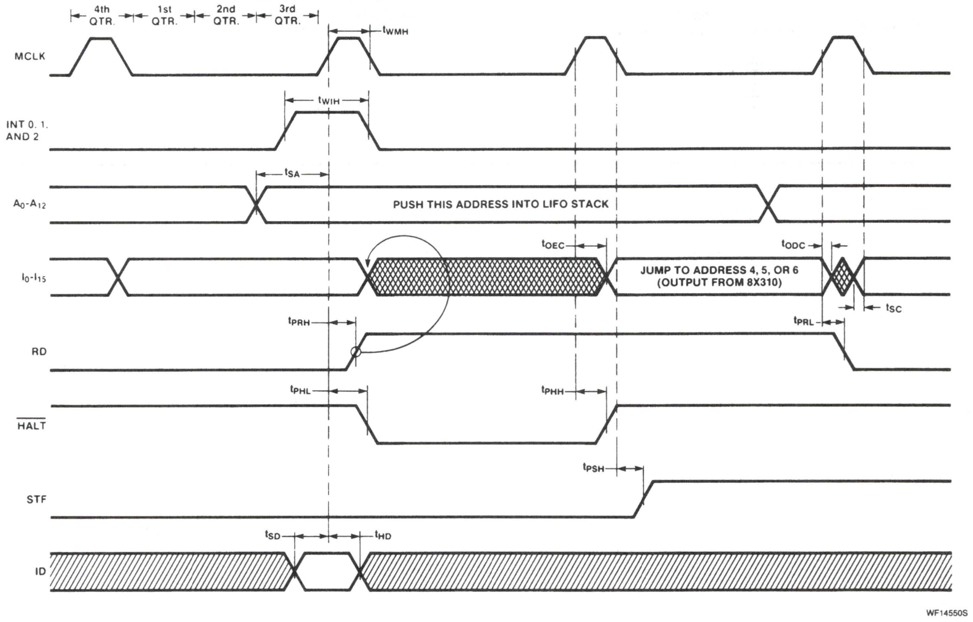
NOTES:

1. V_{MEAS} = 1.5V for all input signals and RD, STF, and HALT outputs.
2. For $I_0 - I_{15}$ (t_{OEC}): V_{MEAS} = 1.5V
   For $I_0 - I_{15}$ (t_{ODC}): V_{MEAS} (t_{PLZ}) = V_{OL} + 0.5V
   V_{MEAS} (t_{PHZ}) = V_{OH} − 0.5V

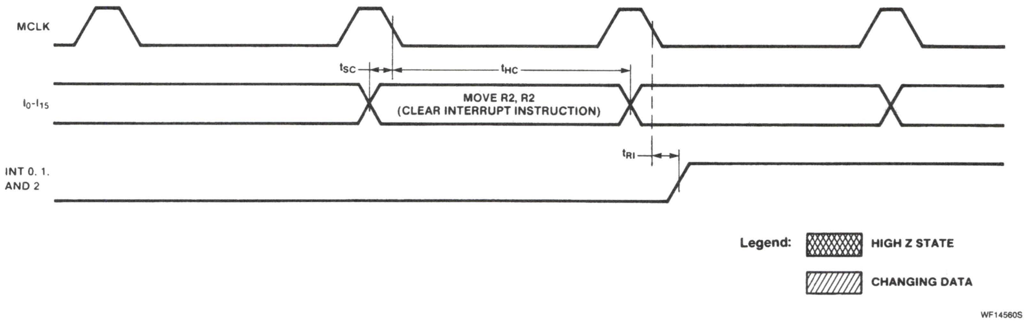
TIMING DIAGRAMS


MCLK
INT 0, 1 AND 2
tHIL
tSIH
tWIL
tWIH
WF14540S

Interrupt Request Timing

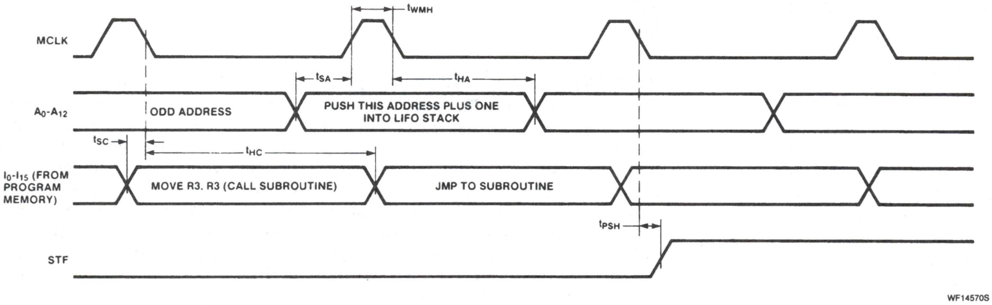

4th QTR.
1st QTR.
2nd QTR.
3rd QTR.
tWMH
MCLK
INT 0, 1, AND 2
tWIH
A0-A12
tSA
PUSH THIS ADDRESS INTO LIFO STACK
I0-I15
tOEC
tODC
JUMP TO ADDRESS 4, 5, OR 6 (OUTPUT FROM 8X310)
tSC
RD
tPRH
tPRL
HALT
tPHL
tPHH
STF
tPSH
ID
tSD
tHD
WF14550S

Interrupt Service Timing

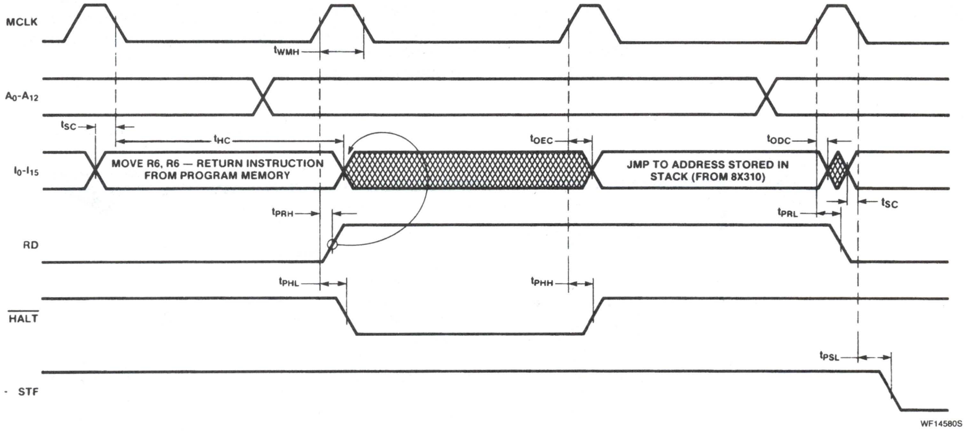

MCLK
I0-I15
tSC
tHC
MOVE R2, R2 (CLEAR INTERRUPT INSTRUCTION)
INT 0, 1, AND 2
tRI
Legend:
HIGH Z STATE
CHANGING DATA
WF14560S

Clear Interrupt Instruction Timing

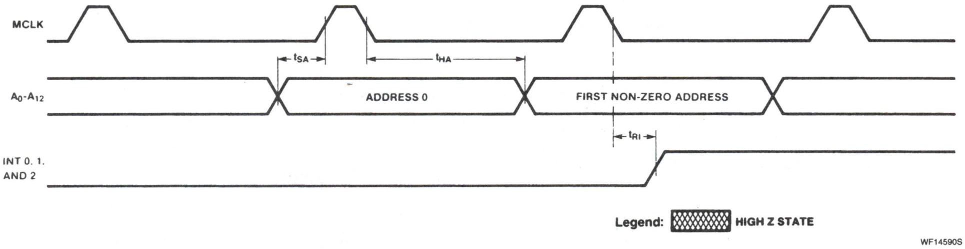
TIMING DIAGRAMS (Continued)


MCLK
t_{WMH}
t_{SA}
t_{HA}
A_0-A_{12}
ODD ADDRESS
PUSH THIS ADDRESS PLUS ONE INTO LIFO STACK
t_{SC}
t_{HC}
I_0-I_{15} (FROM PROGRAM MEMORY)
MOVE R3, R3 (CALL SUBROUTINE)
JMP TO SUBROUTINE
t_{PSH}
STF
WF14570S

Call Subrutine Timing


MCLK
t_{WMH}
A_0-A_{12}
t_{SC}
t_{HC}
t_{OEC}
t_{ODC}
I_0-I_{15}
MOVE R6, R6 — RETURN INSTRUCTION FROM PROGRAM MEMORY
JMP TO ADDRESS STORED IN STACK (FROM 8X310)
t_{SC}
t_{PRH}
t_{PRL}
RD
t_{PHL}
t_{PHH}
$\overline{HALT}$
t_{PSL}
STF
WF14580S

Return Timing (From Interrupt or Subroutine)


MCLK
t_{SA}
t_{HA}
A_0-A_{12}
ADDRESS 0
FIRST NON-ZERO ADDRESS
t_{RI}
INT 0, 1, AND 2
Legend: ▨ HIGH Z STATE
WF14590S

Reset Timing

Signetics

8X320
Bus Interface Register Array

Product Specification

Microprocessor Products

FEATURES
- 16-byte/2-port interface
- 8- or 16-bit primary port (Host) interface (User selectable)
- 8-bit secondary port interface
- Two 8-bit flag registers (handshake control)
- DMA or programmed I/O operation
- Two 3-State bidirectional ports
- Secondary port is bus compatible with 8X305
- Single 5V supply
- 40-pin package

ARCHITECTURAL OVERVIEW
The Signetics 8X320 Bus Interface Register Array (Figure 1) is a dual-port RAM memory designed for use between a host processor and a peripheral processor. Specifically, the register array provides a convenient and economical interface between the 8X305 Microcontroller (secondary port) and User's Host System (primary port); the host can be almost any bus-oriented device — another processor, a minicomputer, or a mainframe computer. The host has 8-bit (byte) or 16-bit (word) access to the primary port; data can be read-from or written-into any memory location as determined by the primary-port address and control lines. The secondary port (8X305 bus) consists of eight input/output lines and four bus control lines. To implement the secondary-port interface, an 8-bit memory location is addressed during one machine cycle and, during another cycle, data is read or written under control of the secondary (8X305) processor. Both primary and secondary ports feature 3-State outputs and both ports are bidirectional.

Besides the convenience and economy of a two-port memory, the array also provides simple handshake control via two 8-bit flag registers, logic to facilitate DMA transfers, and a write-protect feature for the primary port in both byte and word modes of operation.

BLOCK DIAGRAM

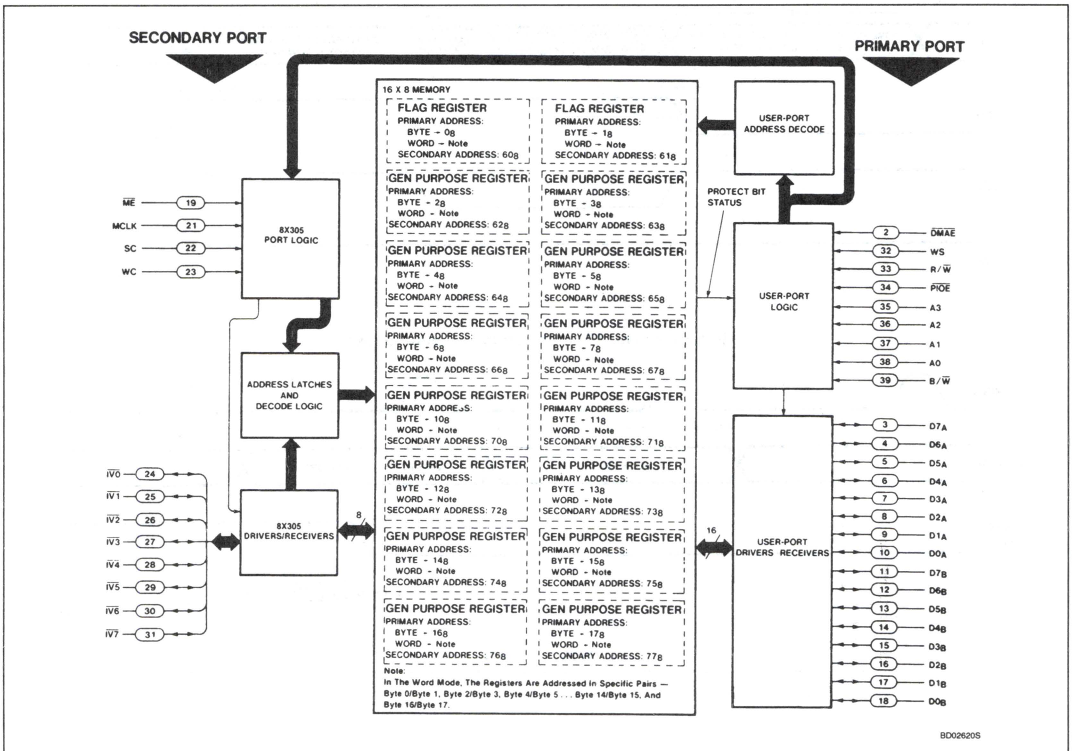
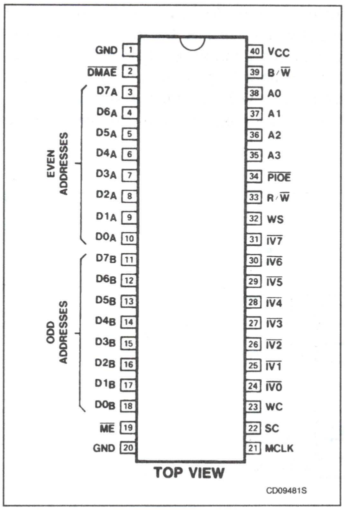

SECONDARY PORT
PRIMARY PORT
16 X 8 MEMORY
FLAG REGISTER PRIMARY ADDRESS: BYTE - 0₈ WORD - Note SECONDARY ADDRESS: 60₈
FLAG REGISTER PRIMARY ADDRESS: BYTE - 1₈ WORD - Note SECONDARY ADDRESS: 61₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 2₈ WORD - Note SECONDARY ADDRESS: 62₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 3₈ WORD - Note SECONDARY ADDRESS: 63₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 4₈ WORD - Note SECONDARY ADDRESS: 64₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 5₈ WORD - Note SECONDARY ADDRESS: 65₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 6₈ WORD - Note SECONDARY ADDRESS: 66₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 7₈ WORD - Note SECONDARY ADDRESS: 67₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 10₈ WORD - Note SECONDARY ADDRESS: 70₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 11₈ WORD - Note SECONDARY ADDRESS: 71₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 12₈ WORD - Note SECONDARY ADDRESS: 72₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 13₈ WORD - Note SECONDARY ADDRESS: 73₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 14₈ WORD - Note SECONDARY ADDRESS: 74₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 15₈ WORD - Note SECONDARY ADDRESS: 75₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 16₈ WORD - Note SECONDARY ADDRESS: 76₈
GEN PURPOSE REGISTER PRIMARY ADDRESS: BYTE - 17₈ WORD - Note SECONDARY ADDRESS: 77₈
Note: In The Word Mode, The Registers Are Addressed In Specific Pairs — Byte 0/Byte 1, Byte 2/Byte 3, Byte 4/Byte 5 . . . Byte 14/Byte 15, And Byte 16/Byte 17.
ME 19
MCLK 21
SC 22
WC 23
8X305 PORT LOGIC
ADDRESS LATCHES AND DECODE LOGIC
8X305 DRIVERS/RECEIVERS
8
IV0 24
IV1 25
IV2 26
IV3 27
IV4 28
IV5 29
IV6 30
IV7 31
USER-PORT ADDRESS DECODE
PROTECT BIT STATUS
USER-PORT LOGIC
2 DMAE
32 WS
33 R/W
34 PIOE
35 A3
36 A2
37 A1
38 A0
39 B/W
16
USER-PORT DRIVERS RECEIVERS
3 D7A
4 D6A
5 D5A
6 D4A
7 D3A
8 D2A
9 D1A
10 D0A
11 D7B
12 D6B
13 D5B
14 D4B
15 D3B
16 D2B
17 D1B
18 D0B
BD02620S

Figure 1. Block Diagram of 8X320 Bus Interface Register Array

Bus Interface Register Array

8X320

PIN CONFIGURATION


GND 1
$\overline{DMAE}$ 2
D7A 3
D6A 4
D5A 5
D4A 6
D3A 7
D2A 8
D1A 9
D0A 10
D7B 11
D6B 12
D5B 13
D4B 14
D3B 15
D2B 16
D1B 17
D0B 18
$\overline{ME}$ 19
GND 20
EVEN ADDRESSES
ODD ADDRESSES
40 VCC
39 B/$\overline{W}$
38 A0
37 A1
36 A2
35 A3
34 $\overline{PIOE}$
33 R/$\overline{W}$
32 WS
31 $\overline{IV7}$
30 $\overline{IV6}$
29 $\overline{IV5}$
28 $\overline{IV4}$
27 $\overline{IV3}$
26 $\overline{IV2}$
25 $\overline{IV1}$
24 $\overline{IV0}$
23 WC
22 SC
21 MCLK
TOP VIEW
CD09481S

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
40-Pin plastic DIP	N8X320N
40-Pin ceramic DIP	N8X320I

PIN DESCRIPTION

PIN NO.	IDENTIFIER	FUNCTION
1, 20	GND	**Ground** — Circuit ground.
2	$\overline{DMAE}$	**Direct Memory Access Enable** — Enables primary port to facilitate DMA transfers; does not affect secondary port.
3 – 18	$D0_A - D7_A/$ $D0_B - D7_B$	**Primary Data Port** — Sixteen 3-State lines used for data transfers to-and-from the primary data port; most significant bit is $D0_B$ and least significant bit is $D7_A$.
19	$\overline{ME}$	**Master Enable** — Enables secondary port when active Low ($\overline{ME}$).
21	MCLK	**Master Clock** — When MCLK is High, and 8X320 is enabled ($\overline{ME}$ = Low), a register location may be either selected or written-into under control of SC and WC.
22	SC	**Select Command** — With SC High, WC Low, MCLK High and $\overline{ME}$ Low, data on $\overline{IV0}$ through $\overline{IV7}$ is interpreted as an address. If any one of the 16 register addresses ($60_8 - 77_8$) matches that on the I/O ($\overline{IV}$) bus, that particular register is selected and remains selected until another address on the same bank (i.e. $\overline{ME}$ = low) is output on the I/O bus — at which time, the old register is deselected and a new register may or may not be selected.
23	WC	**Write Command** — With WC High, SC Low, MCLK High, and $\overline{ME}$ Low, the selected register stores contents of $\overline{IV0} - \overline{IV7}$ as data.
24 – 31	$\overline{IV0} - \overline{IV7}$	**Secondary Data Port** — Eight 3-State lines used to transfer data or I/O address to-and-from the secondary data port; most significant bit is $\overline{IV0}$ and least significant bit is $\overline{IV7}$.
32	WS	**Write Strobe** — When active High, data appearing at the primary port ($D0_A - D7_A/D0_B - D7_B$) is stored in the register array if the primary port is in the *write* mode.
33	R/$\overline{W}$	**Read/Write Control** — When this signal is High, primary port is in *read* mode; when signal is Low, primary port is in *write* mode.
34	$\overline{PIOE}$	**Programmed I/O Enable** — When active Low, primary port operates in programmed input/output mode with register to be read-from or written-into selected by A0 – A3.
35 – 38	A0 – A3	**Primary Port Address Select** — Selects register or register-pair that primary port is to read-from or write-into. Most significant bit is A3; least significant bit is A0.
39	B/$\overline{W}$	**Byte/Word** — When signal is High, the primary port operates in the byte (8-bit) mode; when signal is Low, the primary port operates in the word (16-bit) mode.
40	V_{CC}	**Power** — +5V.

NOTE:
All barred symbols ($\overline{DMAE}$, etc.) denote signals that are asserted (or active) when Low (logical 0); signals that are not barred are asserted in the High state (logical 1).

Bus Interface Register Array

8X320

OPERATING CHARACTERISTICS

Memory Organization

Memory and address correlation for the 16-register array is shown in Figure 2. From the primary port, the sixteen 8-bit registers can be addressed in either byte (8-bit) or word (16-bit) format; in the word mode, the registers are addressed in pairs — $0_8/1_8$, $2_8/3_8$, $4_8/5_8$,...$14_8/15_8$, and $16_8/17_8$. From the secondary port, all registers are addressed in byte format — 60_8 through 77_8. The memory consists of two 8-bit flag registers and fourteen 8-bit general-purpose registers. The flag registers facilitate information transfers between the two ports and, in addition, they protect certain registers from being written into from the primary port.

In either byte or word mode, the write-protect logic, implemented by bits F0 and F1 of register 60_8, inhibits the primary port from writing into addresses 16_8 and 17_8, respectively. Both write-protect bits (F0 and F1) can be read or written from the secondary port; the bits are read-only from the primary port.

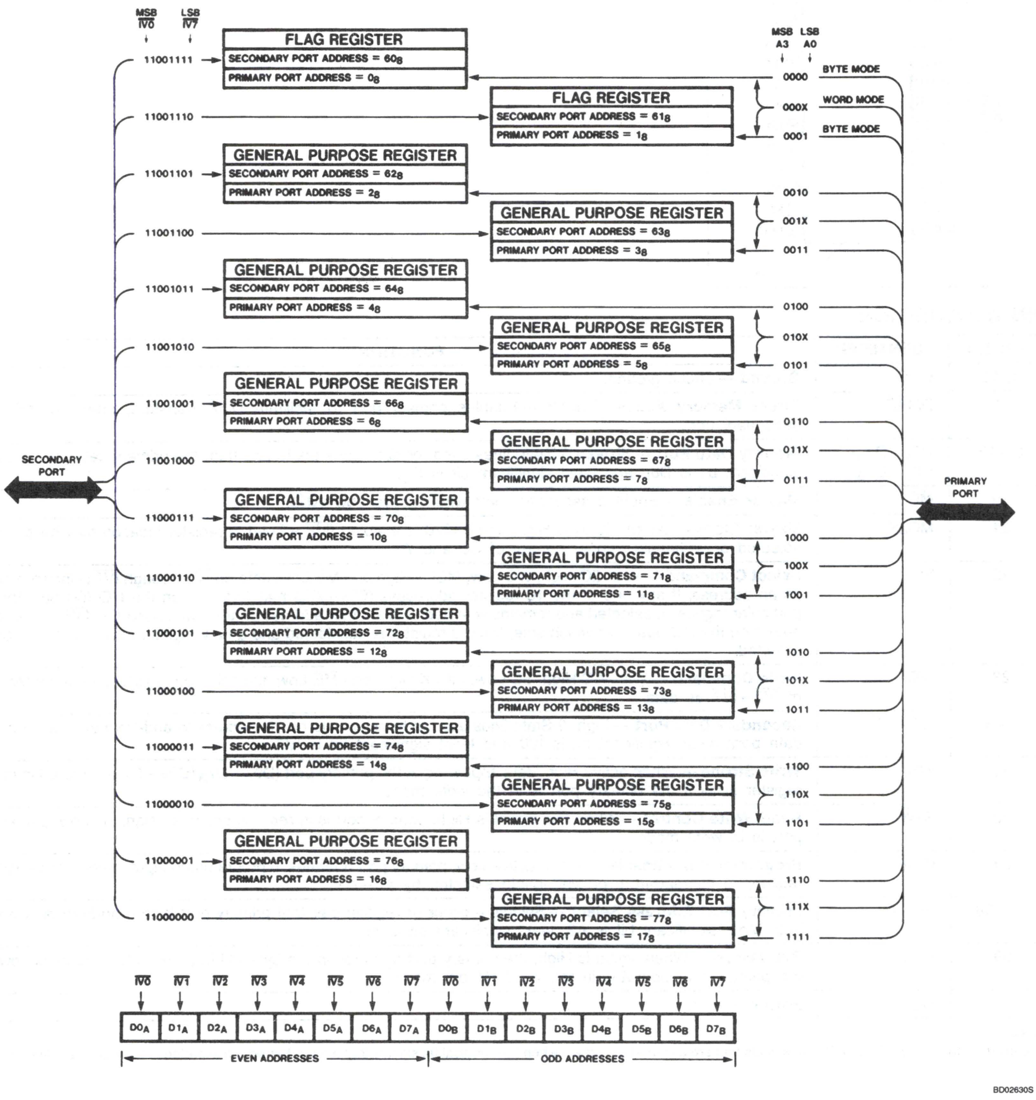

MSB IV0
LSB IV7
11001111
FLAG REGISTER
SECONDARY PORT ADDRESS = 60₈
PRIMARY PORT ADDRESS = 0₈
11001110
FLAG REGISTER
SECONDARY PORT ADDRESS = 61₈
PRIMARY PORT ADDRESS = 1₈
11001101
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 62₈
PRIMARY PORT ADDRESS = 2₈
11001100
GENERAL PURPOSE REGISTER
SECONDARY PORT ADDRESS = 63₈
PRIMARY PORT ADDRESS = 3₈
11001011
SECONDARY PORT ADDRESS = 64₈
PRIMARY PORT ADDRESS = 4₈
11001010
SECONDARY PORT ADDRESS = 65₈
PRIMARY PORT ADDRESS = 5₈
11001001
SECONDARY PORT ADDRESS = 66₈
PRIMARY PORT ADDRESS = 6₈
11001000
SECONDARY PORT ADDRESS = 67₈
PRIMARY PORT ADDRESS = 7₈
11000111
SECONDARY PORT ADDRESS = 70₈
PRIMARY PORT ADDRESS = 10₈
11000110
SECONDARY PORT ADDRESS = 71₈
PRIMARY PORT ADDRESS = 11₈
11000101
SECONDARY PORT ADDRESS = 72₈
PRIMARY PORT ADDRESS = 12₈
11000100
SECONDARY PORT ADDRESS = 73₈
PRIMARY PORT ADDRESS = 13₈
11000011
SECONDARY PORT ADDRESS = 74₈
PRIMARY PORT ADDRESS = 14₈
11000010
SECONDARY PORT ADDRESS = 75₈
PRIMARY PORT ADDRESS = 15₈
11000001
SECONDARY PORT ADDRESS = 76₈
PRIMARY PORT ADDRESS = 16₈
11000000
SECONDARY PORT ADDRESS = 77₈
PRIMARY PORT ADDRESS = 17₈
SECONDARY PORT
PRIMARY PORT
MSB A3
LSB A0
0000
BYTE MODE
000X
WORD MODE
0001
BYTE MODE
0010
001X
0011
0100
010X
0101
0110
011X
0111
1000
100X
1001
1010
101X
1011
1100
110X
1101
1110
111X
1111
IV0 IV1 IV2 IV3 IV4 IV5 IV6 IV7 IV0 IV1 IV2 IV3 IV4 IV5 IV6 IV7
D0A D1A D2A D3A D4A D5A D6A D7A D0B D1B D2B D3B D4B D5B D6B D7B
EVEN ADDRESSES
ODD ADDRESSES
BD02630S

Figure 2. Memory and Address Organization for the 8X320

Table 1. Control of the Two Flag Register

FLAG REGISTERS	60_8 (0_8)	F2	F3	F4	F5	F6	F7								
	61_8 (1_8)							F0	F1	F2	F3	F4	F5	F6	F7
OCTAL ADDRESS OF CONTROLLING BYTE	Primary	2	3	4	5	6	7	10	11	12	13	14	15	16	17
	Secondary	62	63	64	65	66	67	70	71	72	73	74	75	76	77

Table 2. Relationship Between Flag Register Bits and Those of Primary and Secondary Ports.

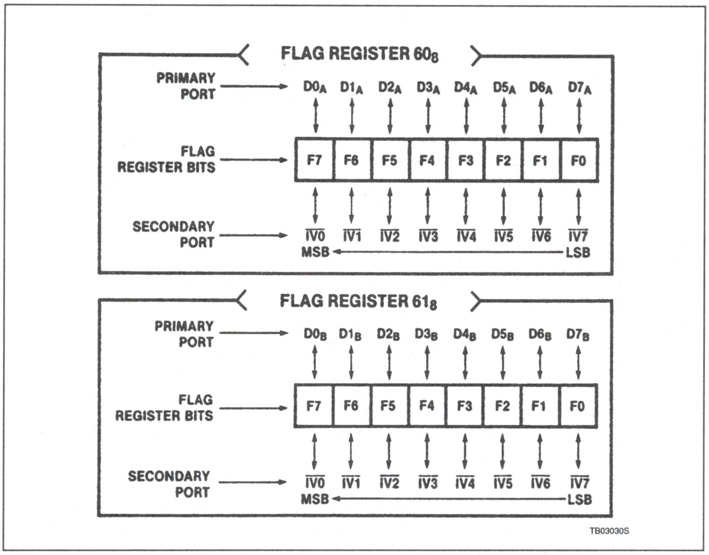

FLAG REGISTER 60_8
PRIMARY PORT
$D0_A$ $D1_A$ $D2_A$ $D3_A$ $D4_A$ $D5_A$ $D6_A$ $D7_A$
FLAG REGISTER BITS
F7 F6 F5 F4 F3 F2 F1 F0
SECONDARY PORT
$\overline{IV0}$ $\overline{IV1}$ $\overline{IV2}$ $\overline{IV3}$ $\overline{IV4}$ $\overline{IV5}$ $\overline{IV6}$ $\overline{IV7}$
MSB
LSB
FLAG REGISTER 61_8
PRIMARY PORT
$D0_B$ $D1_B$ $D2_B$ $D3_B$ $D4_B$ $D5_B$ $D6_B$ $D7_B$
FLAG REGISTER BITS
F7 F6 F5 F4 F3 F2 F1 F0
SECONDARY PORT
$\overline{IV0}$ $\overline{IV1}$ $\overline{IV2}$ $\overline{IV3}$ $\overline{IV4}$ $\overline{IV5}$ $\overline{IV6}$ $\overline{IV7}$
MSB
LSB
TB03030S

Table 3. Mode Control of Primary Port

MODE	$\overline{\text{PIOE}}$	$\overline{\text{DMAE}}$
Disabled (output)	1	1
Programmed I/O	0	1
DMA	X	0

NOTE:
X = Don't Care

Table 4. Primary Port Operating in Programmed I/O Mode

MODE	$B/\overline{W}$	A0	$D0_A - D7_A$ (EVEN ADDRESSES)	$D0_B - D7_B$ (ODD ADDRESSES)
Read	0 (Word)	X	Stored Data	Stored Data
Read	1 (Byte)	0	Stored Data	HI-Z
Read	1 (Byte)	1	HI-Z	Stored Data
Write	0 (Word)	X	Write	Write
Write	1 (Byte)	0	Write	No Change
Write	1 (Byte)	1	No Change	Write

NOTE:
X = Don't Care

As shown in Table 1, flag bits F2 through F7 of 60_8 and F0 through F7 of 61_8 are controlled by the fourteen general-purpose registers. When any one of these registers is written into by either port, the corresponding flag bit for that register is automatically set by internal logic of the 8X320. When information is read from any register, the corresponding flag bit must be reset by user software. Except for the write-protect bits, all other flag bits can be read or reset from the primary or the secondary port. Table 2 shows the relationship between bits of the flag registers and bits of the primary and secondary ports.

FUNCTION AND CONTROL OF PRIMARY PORT

The primary port provides an 8-bit (byte) or 16-bit (word) interface between the 16-byte memory and the user's host system. If the host is an 8-bit system (or 16-bit system operating in Byte mode), the sixteen bidirectional I/O lines must be tied together ($D0_A$ to $D0_B$, $D1_A$ to $D1_B$, ... and $D7_A$ to $D7_B$); when data is input or output on $D0_A$ through $D7_A$, the remaining eight lines ($D0_B$ through $D7_B$) are high-Z and vice-versa.

Other than the Byte/$\overline{\text{Word}}$ control line, specific operating characteristics of the primary port are controlled by two signals — $\overline{\text{PIOE}}$ (Programmed I/O Enable) and $\overline{\text{DMAE}}$ (Direct Memory Access Enable). When $\overline{\text{PIOE}}$ is active (low) and $\overline{\text{DMAE}}$ is inactive (high), the primary port operates in the programmed I/O mode — refer to Table 3; in this mode of operation, the register to be read-from or written-into is determined by four address lines (A0 through A3) and the Byte/$\overline{\text{Word}}$ control line — see Figure 2 and Table 4. In the DMA mode of operation, A1, A2, and A3 are not used; data is read-from or written-into preassigned registers: bytes 16_8 (76_8) and 17_8 (77_8) for the byte mode of operation and bytes 14_8 (74_8)/15_8 (75_8) and 16_8 (76_8)/17_8 (77_8) for the word mode of operation. In both cases, switching between bytes 16_8 and 17_8 in the byte mode and 14_8/15_8 and 16_8/17_8 in the word mode is controlled by A0 (the least significant address bit). Refer to Table 5.

Table 4 defines programmed I/O operation of the primary port in terms of read/$\overline{\text{write}}$ functions and Byte/$\overline{\text{Word}}$ control. In the byte mode, data is read-from or written-into the even addresses (0_8, 2_8, 4_8, 6_8, 10_8, 12_8, 14_8, and 16_8) via data lines $D0_A$ through $D7_A$; data is read-from or written-into odd addresses (1_8, 3_8, 5_8, 7_8, 11_8, 13_8, 15_8, and 17_8) via data lines $D0_B$ through $D7_B$. When A0 is low (logical 0), even addresses are selected and when A0 is high (logical 1), odd addresses are selected; thus, A0 is the LSB of a 4-bit address. In the word mode, the state of A0 is irrelevant, since both the odd and even bytes are, simultaneously, read-from or written-into; thus, a register pair is selected by a 3-bit address, A1 being the LSB.

In the DMA mode of operation with $\overline{\text{DMAE}}$ set to 0 and other conditions satisfied, data is directly transferred to-or-from specified memory locations under control of Byte/$\overline{\text{Word}}$, R/$\overline{\text{W}}$, and A0. The state of the Byte/$\overline{\text{Word}}$ control line determines whether the data word is 8 bits or 16 bits. The A0 address line correlates eight of the sixteen data lines ($D0_A - D7_A$ or $D0_B - D7_B$) with the proper byte/word location. Thus, in the word mode, the exchange of data between the memory and the primary port occurs via $D0_A - D7_A$ for bytes 14_8 and 16_8 and via $D0_B - D7_B$ for bytes 15_8 and 17_8. The byte mode of operation is similar, except that the unused eight lines are 3-Stated.

Table 5. DMA Operation of the Primary Port

MODE	BYTE/WORD	A0	$D0_A - D7_A$	$D0_B - D7_B$
Read	0 (Word)	0	Data stored in byte 14_8	Data stored in byte 15_8
Read	0 (Word)	1	Data stored in byte 16_8	Data stored in byte 17_8
Read	1 (Byte)	0	Data stored in byte 16_8	HI-Z
Read	1 (Byte)	1	HI-Z	Data stored in byte 17_8
Write	0 (Word)	0	Write to byte 14_8	Write to byte 15_8
Write	0 (Word)	1	Write to byte 16_8	Write to byte 17_8
Write	1 (Byte)	0	Write to byte 16_8	HI-Z
Write	1 (Byte)	1	HI-Z	Write to byte 17_8

FUNCTION AND CONTROL OF SECONDARY PORT

The secondary port provides an 8-bit interface between the sixteen memory registers and the 8X305 (or other processor). As shown in Table 6, the secondary-port interface is controlled by five input signals and a status latch. The status latch is set when SC is high (MCLK high/$\overline{\text{ME}}$ low) and a valid memory address ($60_8 - 77_8$) is presented to the 8X320 via the secondary data port ($\overline{\text{IV0}} - \overline{\text{IV7}}$). The latch is cleared by internal logic when an invalid memory address is presented at the secondary port. In all read/write operations from the secondary port, the status latch acts like a master enable; data can be transferred only if the status latch is set.

Table 6. Functional Control of Secondary Port

$\overline{\text{ME}}$	SC[1]	WC[1]	MCLK	R/$\overline{\text{W}}$	STATUS LATCH	FUNCTION OF SECONDARY BUS
L	L	L	X	X	Set	Output data from 8X320 memory to 8X305.
L	L	H	H	H	Set	Data from 8X305 is input and written-into a previously-selected memory location of the 8X320 (Note 2).
L	L	H	H	L	Set	With the primary port in the write mode (R/$\overline{\text{W}}$ = 0), the secondary port is overridden and cannot write to the same register addressed by the primary port; however, the register addressed by the primary port can be read and any other register can be read-from or written-into from the secondary port (Note 2).
L	H	L	H	X	X	Data transmitted to the secondary port via the $\overline{\text{IV}}$ bus is interpreted as an address; if address is within range of $60_8 - 77_8$ the memory status latch is subsequently set.
L	L	H	L	X	X	Inactive
L	H	L	L	X	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

NOTES:

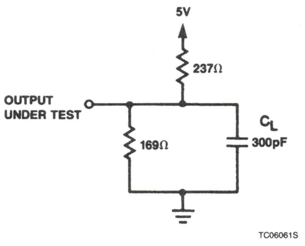
1. The SC and WC lines should never both be high at the same time; the 8X305 processor never generates this condition.
2. During read or write operations, the same register can be simultaneously addressed from either port. For any write operation by both ports on the same register, the primary port has priority; other than this, the 8X320 does not indicate error conditions or resolve conflicts.
3. X = Don't Care.

DC ELECTRICAL CHARACTERISTICS $0°C \leqslant T_A \leqslant 70°C$; $4.75V \leqslant 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS[1, 2]	LIMITS			UNIT
			Min	Typ	Max	
V_{CC}	Supply voltage		4.75	5	5.25	V
$V_{IN\ (L)}$	Low-level input voltage				0.8	V
$V_{IN\ (H)}$	High-level input voltage		2			V
V_{OL}	Low-level output voltage	V_{CC} = 4.75V; I_{OL} = 16mA			0.55	V
V_{OH}	High-level output voltage	V_{CC} = 4.75V; I_{OH} = −3mA	2.4			V
V_{CL}	Input clamp voltage	I_I = −10mA			−1.5	V
I_{CC}	Supply current	V_{CC} = 5.25V (Both ports high-Z)			270	mA
I_{OS}	Short circuit output current[3]	V_{CC} = 5.25V	−20		−100	mA
$I_{IN\ (L)}$	WC, MCLK, SC, & $\overline{ME}$	V_{CC} = 5.25V; V_{IL} = 0.50V			−1	mA
$I_{IN\ (L)}$	B/$\overline{W}$	V_{CC} = 5.25V; V_{IL} = 0.50V			−1.6	mA
$I_{IN\ (L)}$	A0 – A3	V_{CC} = 5.25V; V_{IL} = 0.50V			−1	mA
$I_{IN\ (L)}$	$\overline{DMAE}$	V_{CC} = 5.25V; V_{IL} = 0.5V			−800	μA
$I_{IN\ (L)}$	WS, $\overline{PIOE}$, & R/$\overline{W}$	V_{CC} = 5.25V; V_{IL} = 0.5V			−400	μA
$I_{IN\ (L)}$	$\overline{IV0}$ – $\overline{IV7}$	V_{CC} = 5.25V; V_{IL} = 0.5V			−400	μA
$I_{IN\ (L)}$	$D0_A - D7_A/D0_B - D7_B$	V_{CC} = 5.25V; V_{IL} = 0.5V			−400	μA
$I_{IN\ (H)}$	WC, SC, MCLK, & $\overline{ME}$	V_{CC} = 5.25V; V_{IH} = 5.25V			100	μA
$I_{IN\ (H)}$	B/$\overline{W}$	V_{CC} = 5.25V; V_{IH} = 5.25V			240	μA
$I_{IN\ (H)}$	A0	V_{CC} = 5.25V; V_{IH} = 5.25V			120	μA
$I_{IN\ (H)}$	A1 – A3	V_{CC} = 5.25V; V_{IH} = 5.25V			60	μA
$I_{IN\ (H)}$	$\overline{DMAE}$	V_{CC} = 5.25V; V_{IH} = 5.25V			120	μA
$I_{IN\ (H)}$	WS, $\overline{PIOE}$, & R/$\overline{W}$	V_{CC} = 5.25V; V_{IH} = 5.25V			60	μA
$I_{IN\ (H)}$	$\overline{IV0}$ – $\overline{IV7}$ and $D0_A - D7_A/$ $D0_B - D7_B$	V_{CC} = 5.25V; V_{IH} = 5.25V			100	μA

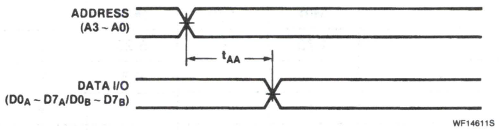
NOTES:
1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. All voltages are measured with respect to ground terminal.
3. Short only one output at a time.

TEST CIRCUIT

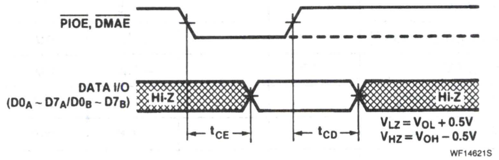


NOTE:
C_L includes jig and probe capacitance.

TIMING DIAGRAMS


ADDRESS (A3 ~ A0)
t_{AA}
DATA I/O ($D0_A$ ~ $D7_A$/$D0_B$ ~ $D7_B$)
WF14611S

Address Access Time


$\overline{\text{PIOE}}$, $\overline{\text{DMAE}}$
DATA I/O ($D0_A$ ~ $D7_A$/$D0_B$ ~ $D7_B$)
Hi-Z
Hi-Z
t_{CE}
t_{CD}
$V_{LZ} = V_{OL} + 0.5V$
$V_{HZ} = V_{OH} - 0.5V$
WF14621S

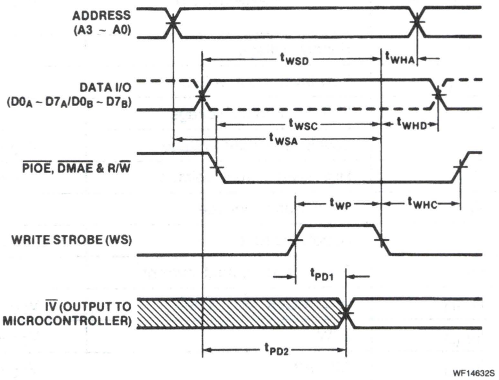
Primary Port Enable/Disable Times

Legend:

CHANGING DATA ▨

Hi-Z ▩

WF14600S


ADDRESS (A3 ~ A0)
t_{WSD}
t_{WHA}
DATA I/O ($D0_A$ ~ $D7_A$/$D0_B$ ~ $D7_B$)
t_{WSC}
t_{WHD}
t_{WSA}
$\overline{\text{PIOE}}$, $\overline{\text{DMAE}}$ & R/$\overline{\text{W}}$
t_{WP}
t_{WHC}
WRITE STROBE (WS)
t_{PD1}
$\overline{\text{IV}}$ (OUTPUT TO MICROCONTROLLER)
t_{PD2}
WF14632S

Write Cycle

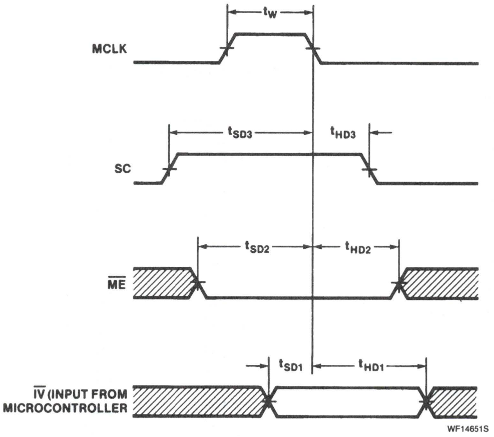
AC CHARACTERISTICS OF PRIMARY PORT

$0°C \leq T_A \leq 70°C$; $4.75V \leq V_{CC} \leq 5.25V$
Loading: See Test Circuit

SYMBOL	PARAMETER	FROM	TO	LIMITS			UNIT
				Min	Typ	Max	
t_{AA}	Address Access Time	A3 – A0	$D0_A - D7_A/D0_B - D7_B$			45	ns
t_{CE}	Primary port enable time	↓ $\overline{\text{PIOE}}$ ↓ $\overline{\text{DMAE}}$	$D0_A - D7_A/D0_B - D7_B$			30	ns
t_{CD}	Primary port disable time	↑ $\overline{\text{PIOE}}$ ↑ $\overline{\text{DMAE}}$	$D0_A - D7_A/D0_B - D7_B$			35	ns
t_{WSA}	Address setup time	A3 – A0	↓ WS	40			ns
t_{WHA}	Address hold time	↓ WS	A3 – A0	0			ns
t_{WSD}	Primary port data setup time	$D0_A - D7_A/$ $D0_B - D7_B$	↓ WS	30			ns
t_{WHD}	Primary port data hold time	↓ WS	$D0_A - D7_A/D0_B - D7_B$	0			ns
t_{WSC}	Write mode control setup time	$\overline{\text{PIOE}}$ $\overline{\text{DMAE}}$ R/$\overline{\text{W}}$	↓ WS ↓ WS ↓ WS	30 40 30			ns ns ns
t_{WHC}	Write mode control hold time	↓ WS	$\overline{\text{PIOE}}$ $\overline{\text{DMAE}}$ R/$\overline{\text{W}}$	10 10 10			ns ns ns
t_{WP}	Write strobe pulse width			25			ns
t_{PD1}[1]	Primary port data delay	$D0_A - D7_A/$ $D0_B - D7_B$	$\overline{\text{IV0}} - \overline{\text{IV7}}$			75	ns
t_{PD2}[2]	Primary port data delay from WS	↑ WS	$\overline{\text{IV0}} - \overline{\text{IV7}}$			75	ns

NOTES:
1. Measurement with Write Strobe set High and the control signals of the secondary port set for output data from the same register.
2. Measurement with primary port data stable and control signals of secondary port set for output data from the same register.

TIMING DIAGRAMS


tW
MCLK
tSD3
tHD3
SC
tSD2
tHD2
$\overline{ME}$
tSD1
tHD1
$\overline{IV}$ (INPUT FROM MICROCONTROLLER
WF14651S

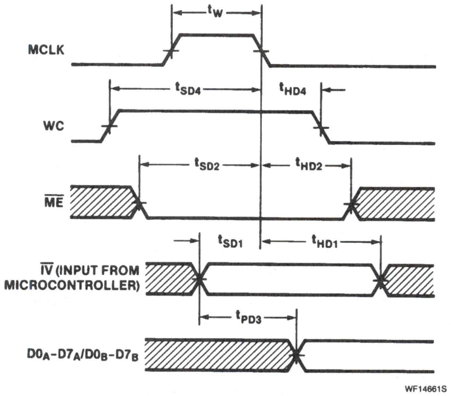
Select Cycle

Legend:

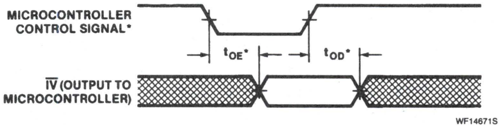
CHANGING DATA ▨

3-STATE ▩

WF14641S


tW
MCLK
tSD4
tHD4
WC
tSD2
tHD2
$\overline{ME}$
tSD1
tHD1
$\overline{IV}$ (INPUT FROM MICROCONTROLLER)
tPD3
D0A-D7A/D0B-D7B
WF14661S

Input Cycle


MICROCONTROLLER CONTROL SIGNAL*
tOE*
tOD*
$\overline{IV}$ (OUTPUT TO MICROCONTROLLER)
WF14671S

Output Cycle

AC CHARACTERISTICS OF SECONDARY PORT

$0°C \leqslant T_A \leqslant 70°C$, $4.75V \leqslant V_{CC} \leqslant 5.25V$
Loading: See Test Circuit

SYMBOL	PARAMETER	FROM	TO	LIMITS			UNIT
				Min	Typ	Max	
t_W	MCLK pulse width			35			ns
t_{SD1}	Data setup time	$\overline{IV0}-\overline{IV7}$	↓ MCLK	35			ns
t_{SD2}	$\overline{ME}$ setup time	$\overline{ME}$	↓ MCLK	35			ns
t_{SD3}	SC setup time	SC	↓ MCLK	35			ns
t_{SD4}	WC setup time	WC	↓ MCLK	35			ns
t_{HD1}	Data hold time	↓ MCLK	$\overline{IV0}-\overline{IV7}$	0			ns
t_{HD2}	$\overline{ME}$ hold time	↓ MCLK	$\overline{ME}$	0			ns
t_{HD3}	SC hold time	↓ MCLK	SC	0			ns
t_{HD4}	WC hold time	↓ MCLK	WC	0			ns
T_{PD3}(Note)	$\overline{IV}$ propagation delay	$\overline{IV}$	$D0_A-D7_A/D0_B-D7_B$			45	ns
T_{OE}	Output enable	$\overline{ME}$, SC, or WC	$\overline{IV0}-\overline{IV7}$			30	ns
T_{OD}	Output disable	$\overline{ME}$, SC, or WC	$\overline{IV0}-\overline{IV7}$			20	ns

NOTE:
Measured with MCLK = High and control signals of the primary port set for output data from the same register.

8X330

Floppy Disk Formatter/ Controller

Product Specification

DESCRIPTION

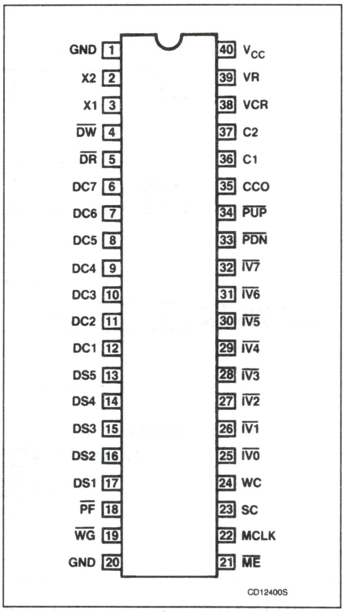
The Signetics 8X330 Floppy Disk Formatter/Controller is a monolithic peripheral device of the 8X300 Family. The chip uses Bipolar-Schottky/1^2L-Technology and some very unique features to provide 8X330 customers with a competitive edge in both simple and complicated disk-controller designs. The competitive advantage is measurable in terms of "systems parts count", "error correction capabilities", and "overall design concepts" that are applications oriented. Except for the following, the 8X330 contains all processing circuits and the required control logic to encode/decode double-density (MFM/ M^{2}FM) and single-density (FM) codes: a crystal; a capacitor; an external transistor acting as a series-pass element for the on-chip voltage regulator; an active low-pass filter; and an optional off chip voltage controlled oscillator (refer to Features and Option). Data-separation and write-precompensation logic are located on the chip, while 16-bytes of scratch-pad RAM are provided for storage of various control/status parameters.

FEATURES

- **Single or double density encoding/decoding**
- **On-chip data separator**
- **Programmable:**
  - **FM, MFM, and M^{2}FM encoding/ decoding**
  - **Preamble Polarity**
  - **Data transfer rate**
  - **Address mark encoding/ decoding**
  - **Sector length**
  - **Output port (7-bits disk command)**
  - **Input port (5-bits disk status)**
- **Write Precompensation with on/ off control**
- **On-chip phase lock loop**
- **CRC generator with software-controlled error correction capabilities**
- **40-pin package**
- **+5V operation**

OPTION: External Voltage Controlled Oscillator (VCO). For critical applications, window margins can be improved by as much as 6% with the use of an external VCO.

PIN CONFIGURATION


GND 1
X2 2
X1 3
DW 4
DR 5
DC7 6
DC6 7
DC5 8
DC4 9
DC3 10
DC2 11
DC1 12
DS5 13
DS4 14
DS3 15
DS2 16
DS1 17
PF 18
WG 19
GND 20
40 VCC
39 VR
38 VCR
37 C2
36 C1
35 CCO
34 PUP
33 PDN
32 IV7
31 IV6
30 IV5
29 IV4
28 IV3
27 IV2
26 IV1
25 IV0
24 WC
23 SC
22 MCLK
21 ME
CD12400S

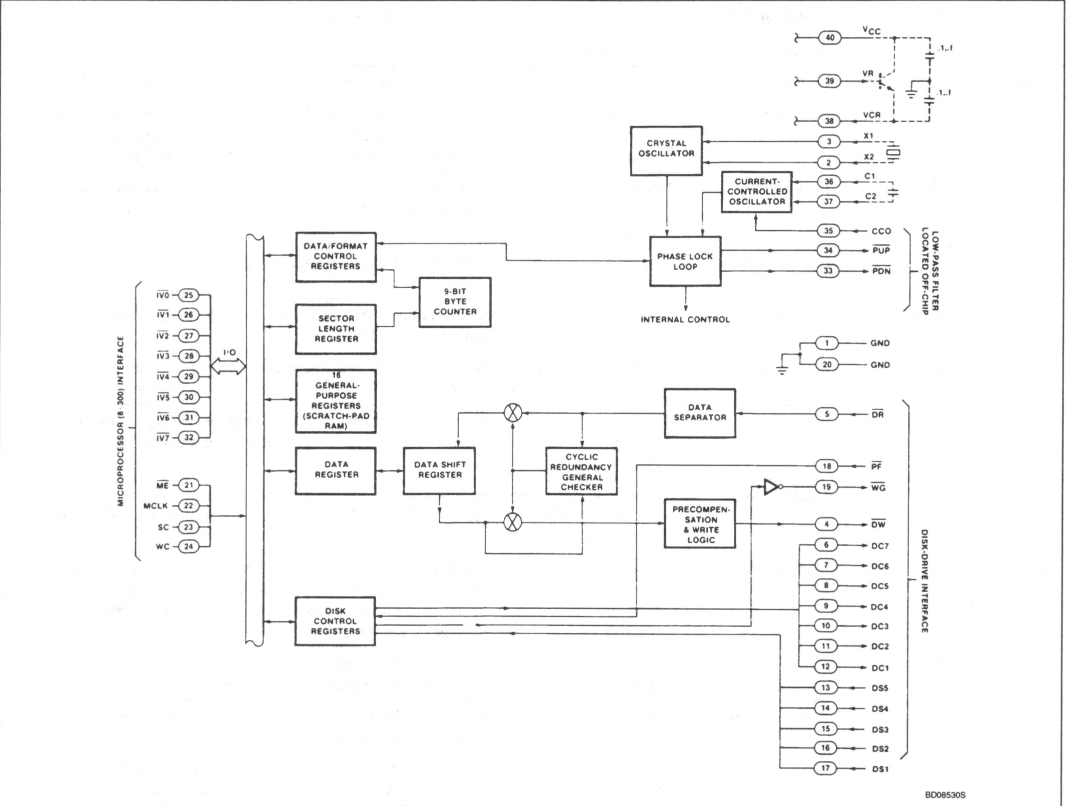
ORDERING INFORMATION

DESCRIPTION	ORDER CODE
40-Pin plastic DIP	N8X330

PIN DESCRIPTION

PIN NO	MNEMONIC	DEFINITION	FUNCTION
1, 20	GND	Ground	Circuit ground
2, 3	X1, X2	Crystal inputs	Inputs from a crystal that determines frequency of an on-chip crystal oscillator.
4	$\overline{\text{DW}}$	Data write	A series of negative-going pulses transmitted to the disk drive. The data write signal produces pulses (with precompensation, if required) for data and clock in accordance with the applicable encoding rules (FM, MFM or M^2FM).
5	$\overline{\text{DR}}$	Data read	Negative-going pulses transmitted from the disk drive to a Schmitt-trigger input of the 8X330; these pulses represent encoded data and clock from the disk media.
6 – 12	DC1 – DC7	Disk commands	Seven outputs from the 8X330 that allow general-purpose control, of one or more disk drives.
13 – 17	DS1 – DS5	Disk status	Five general-purpose Schmitt-trigger inputs from the disk drive (or drives) that provide status information for the 8X300.
18	$\overline{\text{PF}}$	Power fail	Schmitt-trigger input from external logic that is active (low) when the "user-sensed" power supply voltage drops below a predetermined value.
19	$\overline{\text{WG}}$	Write gate	When active (low), this 40mA open-collector output enables writing to the disk media. When $\overline{\text{PF}}$ is low, the write gate is inhibited during periods of power supply uncertainty.
21	$\overline{\text{ME}}$	Master enable	When this input signal is active (low), the 8X330 can be accessed and enabled by the 8X300. (Refer to the $\overline{\text{LB}}$ and $\overline{\text{RB}}$ pinout discriptions of the 8X300 for further detail.)
22	MCLK	Master clock	When active high and with $\overline{\text{ME}}$ in the active-low state, this input signal provides a means whereby the I/O output from the 8X300 is interpreted as an enabling address (provided there is an address match) or as input data (if one of the 8X330 registers has already been selected).
23	SC	Select command	When this signal is active (high), the information output on pins $\overline{\text{IV0}}$ – $\overline{\text{IV7}}$ of the 8X300 is interpreted as an address input by the 8X330.
24	WC	Write command	When this signal is active (high), the information output on pins $\overline{\text{IV0}}$ – $\overline{\text{IV7}}$ of the 8X300 is interpreted as input data by the 8X330.
25 – 32	$\overline{\text{IV0}}$ – $\overline{\text{IV7}}$	Input/output lines	Eight 3-State input/output lines that provide bidirectional data transfers between the 8X300 and the enabled I/O device; $\overline{\text{IV7}}$ is the *Least Significant Bit.*
33	$\overline{\text{PDN}}$	Pump down output	Open-collector output of on-chip phase detector which indicates (by a negative-going, quantized, pulse-width modulated signal) that internal CCO frequency is too high.
34	$\overline{\text{PUP}}$	Pump up output	Open-collector output of on-chip phase detector which indicates (by a negative-going, quantized, pulse-width modulated signal) that internal CCO frequency is too low.
35	CCO	Frequency Control Input for Current-Controlled Oscillator	Variable input current from external low-pass filter that controls the frequency of the oscillator.
36 – 37	C1, C2	Capacitor input terminals	Inputs for capacitor that determines center frequency of the current-controlled oscillator.
38	VCR	Regulated supply voltage	DC voltage input from emitter of external series-pass transistor; this voltage powers internal logic of chip.
39	VR	Reference voltage	Reference voltage output to base of series-pass transistor; this reference controls VCR.
40	V_{CC}	Supply voltage	+5V power.

BLOCK DIAGRAM


MICROPROCESSOR (8X300) INTERFACE
IV0 25
IV1 26
IV2 27
IV3 28
IV4 29
IV5 30
IV6 31
IV7 32
I/O
ME 21
MCLK 22
SC 23
WC 24
DATA/FORMAT CONTROL REGISTERS
SECTOR LENGTH REGISTER
9-BIT BYTE COUNTER
16 GENERAL-PURPOSE REGISTERS (SCRATCH-PAD RAM)
DATA REGISTER
DATA SHIFT REGISTER
CYCLIC REDUNDANCY GENERAL CHECKER
DISK CONTROL REGISTERS
CRYSTAL OSCILLATOR
CURRENT-CONTROLLED OSCILLATOR
PHASE LOCK LOOP
INTERNAL CONTROL
DATA SEPARATOR
PRECOMPENSATION & WRITE LOGIC
40 VCC
39 VR
38 VCR
.1μf
.1μf
3 X1
2 X2
36 C1
37 C2
35 CCO
34 PUP
33 PDN
LOW-PASS FILTER LOCATED OFF-CHIP
1 GND
20 GND
5 DR
18 PF
19 WG
4 DW
6 DC7
7 DC6
8 DC5
9 DC4
10 DC3
11 DC2
12 DC1
13 DS5
14 DS4
15 DS3
16 DS2
17 DS1
DISK-DRIVE INTERFACE
BD08530S

NOTES:

1. Components shown with dotted lines are located off-chip.
2. Certain bits of the status control registers connected to the I/O bus are 'read-only'; this distinction is not shown in the diagram.
3. Greater detail for the more complicated blocks (Phase-Lock Loop, Data Separator, etc.) are shown in other areas of this data sheet.

SYSTEM INTERFACE

A typical flophy disk controller using an 8X300 microcontroller and the 8X330 is shown in Figure 2. The non-shaded portion of this particular configuration can service the command, status, and input/output requirements of two double-sided disk drives and, under software supervision, the system can read/write single-density (FM) or double-density (MFM/M^{2}FM) codes. Interface requirements consist of the 8X300 microcontroller and the two disk drives. All 8X330 control and data registers directly linked to the microprocessor interface (Figure 1) are addressable and appear to the 8X300 as simple I/O ports; a 13-bit address bus and a 16-bit instruction bus provide communications between the 8X300 and up to 8K of microprogram storage.

The disk-drive interface consists of seven output control lines (DC1-DC7), five input status lines (DS1-DS5), a write gate ($\overline{\text{WG}}$), a data-write output ($\overline{\text{DW}}$), and a data-read input ($\overline{\text{DR}}$). The twelve command/status lines are not dedicated; thus, the user can assign system functions to best suit a given application. As shown in Figure 2, all control lines except $\overline{\text{WG}}$ are buffered to accommodate a reasonable distance between the controller and the disk media; the Write Gate, being a 40mA output, requires no buffering.

As shown by the shaded part of Figure 2, the control and status lines can be expanded with peripheral hardware — the 8T32 (in this example) being only one method of implementation. Using this particular technique, one I/O port is totally dedicated to output control, whereas, the other port is totally dedicated to input status. With additional hardware and supporting software, the disk-drive system can be expanded without limit; however, five or six drives is sufficient for most applications. By using the programmable features of the 8X330, the user can emphasize and prioritize those system parameters that are most important — economics, reliability and/or speed.

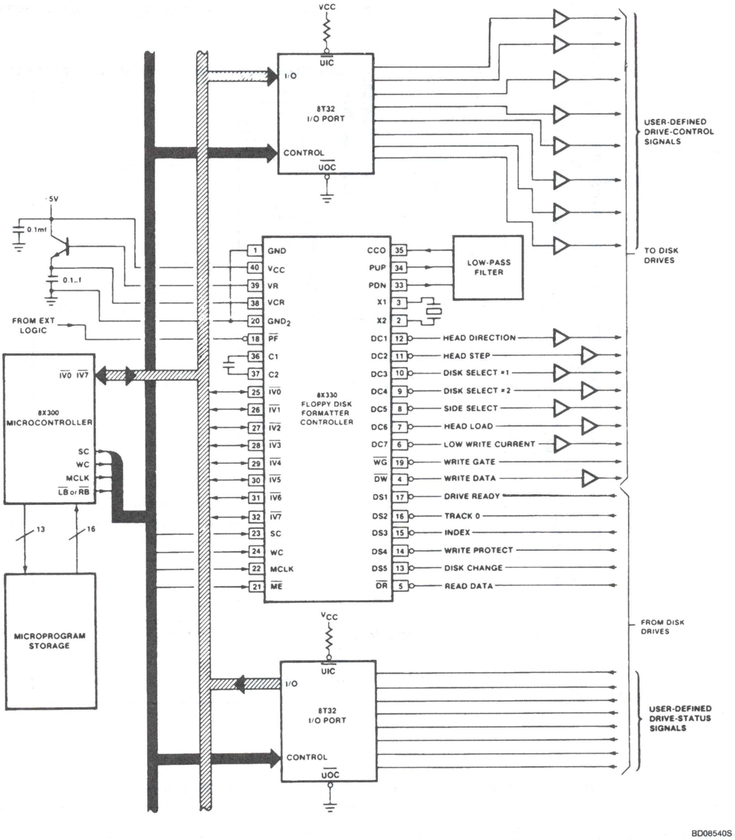

VCC
UIC
I/O
8T32
I/O PORT
CONTROL
UOC
USER-DEFINED
DRIVE-CONTROL
SIGNALS
TO DISK
DRIVES
5V
0.1mf
0.1..f
FROM EXT
LOGIC
IV0 IV7
8X300
MICROCONTROLLER
SC
WC
MCLK
LB or RB
13
16
MICROPROGRAM
STORAGE
1 GND
40 VCC
39 VR
38 VCR
20 GND2
18 PF
36 C1
37 C2
25 IV0
26 IV1
27 IV2
28 IV3
29 IV4
30 IV5
31 IV6
32 IV7
23 SC
24 WC
22 MCLK
21 ME
8X330
FLOPPY DISK
FORMATTER
CONTROLLER
CCO 35
PUP 34
PDN 33
X1 3
X2 2
DC1 12
DC2 11
DC3 10
DC4 9
DC5 8
DC6 7
DC7 6
WG 19
DW 4
DS1 17
DS2 16
DS3 15
DS4 14
DS5 13
DR 5
LOW-PASS
FILTER
HEAD DIRECTION
HEAD STEP
DISK SELECT #1
DISK SELECT #2
SIDE SELECT
HEAD LOAD
LOW WRITE CURRENT
WRITE GATE
WRITE DATA
DRIVE READY
TRACK 0
INDEX
WRITE PROTECT
DISK CHANGE
READ DATA
FROM DISK
DRIVES
VCC
UIC
I/O
8T32
I/O PORT
CONTROL
UOC
USER-DEFINED
DRIVE-STATUS
SIGNALS
BD08540S

Figure 2. Typical Interface Using an 8X300 Microcontroller

FUNCTIONAL OPERATION

As shown in Figure 2, the interface between the 8X300 and 8X330 consists of twelve lines — $\overline{IV0}$-$\overline{IV7}$, SC, WC, MCLK, and $\overline{ME}$; the Master Enable ($\overline{ME}$) input (pin 21) is driven from either the $\overline{LB}$ (Left Bank) or $\overline{RB}$ (Right Bank) output of the 8X300. An expanded view of this interface is shown in Figure 3 and, as indicated, the 8X330 appears as a number of addressable registers (110_8-127_8 and 132_8-137_8) under input/output control of the 8X300. These registers are used for general-purpose storage, data-transfer operations, disk commands, disk status, and various control functions. Design-oriented information for these registers and other data-processing/logic functions of the 8X330 are described in the paragraphs that follow; in all of these registers, bit 0 is the Most Significant Bit (MSB).

NOTE:

When power is first applied to the 8X330, the Disk Command lines (DC1-DC7), the Write Gate ($\overline{WG}$) output, and contents of Command/Status Register #2 (CSR #2) are set to 1 (high). The wakeup state of all other bits is undefined.

General-Purpose Register File

This general-purpose (scratch pad) memory is directly accessible by the 8X300 and is used to store system variables such as track address, sector address and other necessary parameters. The sixteen 8-bit registers (110_8-127_8) provide sufficient on-chip memory to accommodate a minimum of two disk drives; the maximum number of drives that this non-dedicated memory file can support depends on several factors — system configuration, reliability requirements, economic constraints, and so on. Because of the on-chip file, all other system memory can be dedicated to the purpose of handling data to-and-from the disk media.

Floppy Disk Formatter/Controller 8X330

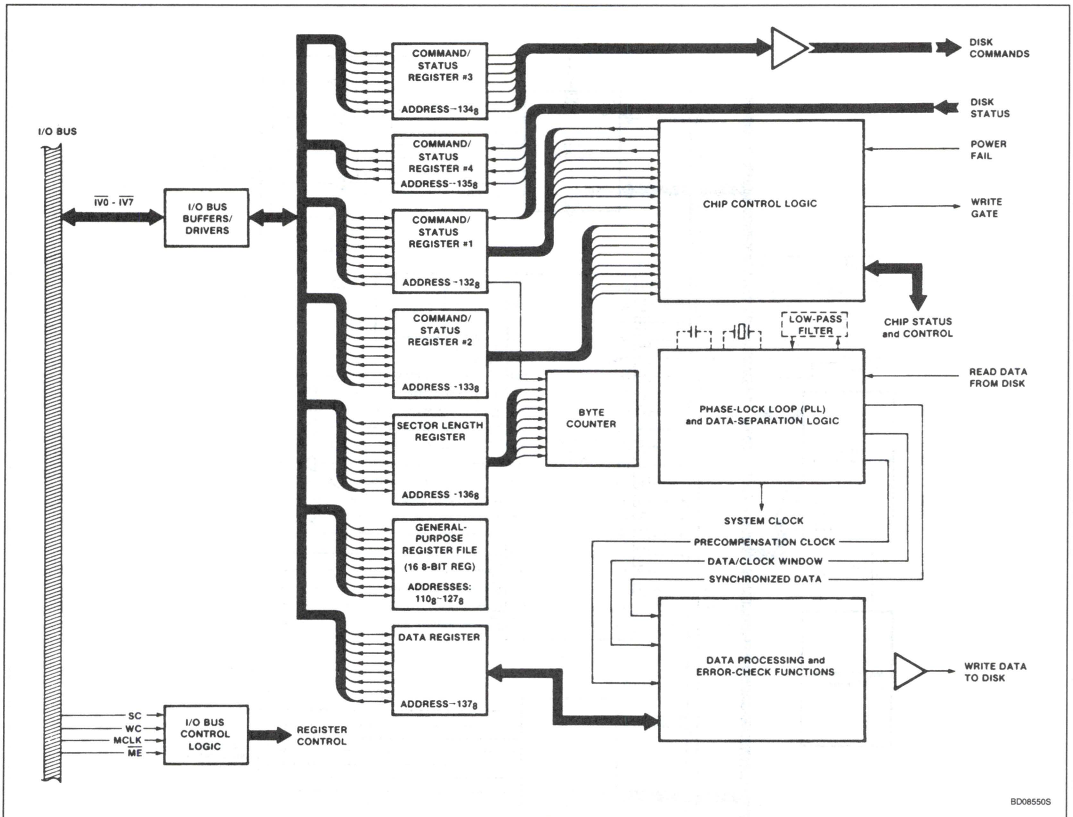

DISK COMMANDS
DISK STATUS
I/O BUS
$\overline{IV0}$ - $\overline{IV7}$
I/O BUS BUFFERS/ DRIVERS
COMMAND/ STATUS REGISTER #3
ADDRESS–134_8
COMMAND/ STATUS REGISTER #4
ADDRESS–135_8
COMMAND/ STATUS REGISTER #1
ADDRESS –132_8
COMMAND/ STATUS REGISTER #2
ADDRESS –133_8
SECTOR LENGTH REGISTER
ADDRESS –136_8
GENERAL-PURPOSE REGISTER FILE (16 8-BIT REG)
ADDRESSES: 110_8–127_8
DATA REGISTER
ADDRESS–137_8
POWER FAIL
CHIP CONTROL LOGIC
WRITE GATE
CHIP STATUS and CONTROL
LOW-PASS FILTER
READ DATA FROM DISK
BYTE COUNTER
PHASE-LOCK LOOP (PLL) and DATA-SEPARATION LOGIC
SYSTEM CLOCK
PRECOMPENSATION CLOCK
DATA/CLOCK WINDOW
SYNCHRONIZED DATA
DATA PROCESSING and ERROR-CHECK FUNCTIONS
WRITE DATA TO DISK
SC
WC
MCLK
$\overline{ME}$
I/O BUS CONTROL LOGIC
REGISTER CONTROL
BD08550S

Figure 3. An Expanded View of the 8X330/8X300 Interface

Command/Status Register #1 (CSR 1/Address 132_8)

The disk status (read) or disk-command (write) contents of this register are interpreted as follows: unless otherwise indicated, all bits of CSR 1 are read/write from the I/O bus.

Bit 0 (Write Gate Enable)

Enables write gate output ($\overline{WG}$/pin 19) to disk drive(s) — the write gate ($\overline{WG}$) cannot be enabled unless the PF input pin (18) is high. When the WGE bit is set to 0, the $\overline{WG}$ output pin is low (enabled); when WGE is set to 1, the $\overline{WG}$ output is high (disabled). If the PF input goes low while the $\overline{WG}$ output is low, the $\overline{WG}$ output will go high and the Write Gate Enable bit is reset to 1.

Bit 1 (CRC Enable)

When set to 1, permits internal CRC register to compute remainders on the data stream in either read or write modes of operation. When set to 0, the CRC register becomes the source of data. A change in the CRC Enable bit does not become effective until the "next BYTRA flag appears" following the command bit change (refer to description of CSR 1/Bit 6).

Bit 2 (Data Register Control)

When set to 0, contents of data register consists of interleaved data-and-clock bits; starting from the MSB ($\overline{IV0}$) position, register contents are: Clock 1, Data 1, Clock 2, Data 2, Clock 3, Data 3, Clock 4, and Data 4. When writing an address mark, the appropriate data/clock pattern is loaded into the data register by the 8X300. Since each byte of data from the processor becomes an interleaved pattern (4-bits of data and 4-bits of clock) in the 8X330 data register, two bytes from the processor are required to write each full byte of address mark to the drive — eight bit cells with each cell containing a possible data and/or clock transition, or a total of 16-bit positions. When writing address marks, the normal on-chip clock insertion circuitry of the 8X330 is inhibited; thus, the user is free to define any clock/data pattern for the address mark.

When reading address marks, the data register is loaded with data and clock representing four bit cells from the disk media. The information in the data register can then be compared with the expected address mark by the 8X300 on a nibble-by-nibble basis. When the DRC bit is set to 1, the data register contains separated data (no clocks). A state change in this bit does not become effective until the 'next BYTRA flag appears" following the state-change command.

Bit 3 (Sync Enable)

The Sync Enable bit allows the on-chip data separator to obtain bit and byte synchronization; this bit also controls initialization of the CRC Register. With the 8X330 in Read mode and with Bit 3 set to 0, bit synchronization occurs. The Preamble field is assumed to be all "zeroes" or all "ones" as determined by the Preamble Select bit (SCR 2/Bit 4).

When the proper number of preamble bytes, as determined by the disk-control program, have been found, the Sync Enable bit should be changed, under program direction, to a 1. This puts the 8X330 in the Address-Mark search mode. Accordingly, all bits of the CRC Register are preset to 1, the BYte TRAnsfer flag is inhibited, and the 8X330 examines the data stream for an Address Mark. The Address Mark is detected by observing the data and clock bits to find a change in the normal Preamble pattern. Byte synchronization is achieved by assuming that the change occurred in the bit cell determined by two Bit Select bits (CSR 2/Bits 2 and 3).

When the pattern change is found indicating the start of an Address Mark, the 8X330 starts CRC computation and synchronizes BYTRA to the byte boundaries. Note that the 8X330 presumes and Address Mark by finding a change in the preamble pattern; however, it is up to the 8X300 to read the Address Mark and to establish its validity or nonvalidity.

In write mode, setting the Sync Enable bit to 0 presets all bits of the CRC Register to 1. Setting the Sync Enable bit to 1 allows CRC computation to begin at the next byte boundary.

Bit 4 (Load Counter)

When set to 1, transfers 8-bits of data from Sector Length register and 1-bit (MSB) of data from Byte Counter (refer to next description) to 9-bit Byte Counter. Loading of the 9-bit Byte Counter is effective one bit-cell time after the Load Counter bit is set to 1. In both the read and write modes of operation, the Byte Counter is incremented by BYTRA. The Load Counter bit is self-clearing and always returns a 0 when read.

NOTE:
The Load Counter bit must be set one or more instruction cycles *after* setting the Byte Counter MSB, that is, bits 4 and 5 of CSR 1 cannot be set during the same instruction cycle.

Bit 5 (Byte Counter MSB)

This bit is used to set and monitor the state of the ninth (MSB) bit in the Byte Counter; reading this bit always returns the current state of MSB in the Byte Counter. The MSB of the Byte Counter is set to the value of CSR 1/Bit 5 when the Load Counter bit (CSR 1/Bit 4) is asserted — refer to preceding description.

NOTE:
The Byte Counter MSB must be set one or more instruction cycles *before* the Load Counter bits — bits 4 and 5 of CSR 1 cannot be set during the same instructions cycle.

Bit 6 (BYTRA)

During a disk read operation, the BYTRA flag is automatically set to 0 when 8-bits of information are transferred from the Data Shift Register to the Data Register — see Figure 1. During a disk write operation, BYTRA is automatically set to 0 when 8-bits are transferred from the Data Register to the Data Shift Register. BYTRA (a read-only bit) is reset to a 1 when the Data Register (address 137_8) is selected by the user's program. During read/write operations, the 1-to-0 transition of the BYTRA flag increments the Byte Counter to keep count of bytes read or bytes written. All read-only bits of the 8X330 are designed to remain stable during the monitor period; thus, to read a status change of BYTRA, Disk-Status bit, the Byte Counter MSB, or other read-only bit requires a two-instruction loop similar to:

```
TEST    SEL     CSR 1
        NZT     BYTRA,  TEST
```

Bit 7 (Disk Status 1)

Reflects state (0 or 1) of input DS1 (pin 17); this is a user-definable read-only bit.

NOTE:
A high input on any one of the Disk Status lines of the 8X330 is read by the 8X300 program as a logical 1 and a low input on the status lines is read as a logical 0.

Command/Status Register #2 (CSR 2/Address 133_8)

The disk status (read) or disk-command (write) contents of this register are interpreted as follows:

Bit 0 (Precompensation Enable)

This command bit determines whether or not precompensation is applied to the data stream being written onto the disk. When set to 0, precompensation is inhibited. When set to 1 and with double-density encoding, write precompensation is applied to the data/clock bit patterns in the table to the left.

Bit 1 (Read Mode)

When set to 0, the 8X330 reads data from the disk and transfers it to the Data Register; when set to 1, data from the Data Register is transferred to the disk, provided the Write Gate Enable bit (CSR1/Bit 0) is set to 1. With WGE set to 0 and the Read Mode bit set to 1, the current-controlled oscillator is forced to lock onto the crystal oscillator; this technique is used during a data-read operation to ensure rapid acquisition of the disk data.

PRECOMP TIME	DATA/CLOCK PATTERN (IN DATA SHIFT REG)			BIT BEING WRITTEN	BITS ALREADY WRITTEN TO DISK		
2T (Late)	0	1	0	1	0	0	0
2T (Late)	0	1	0	1	0	0	1
2T (Early)	1	0	0	1	0	1	0
2T (Early)	0	0	0	1	0	1	0

NOTE:

where, $T = \frac{1}{\text{crystal frequency}}$ if bit 7 of CSR 2(½F) = 1

$T = \frac{2}{\text{crystal frequency}}$ if bit 7 of CSR 2(½F) = 0

Bits 2, 3 (Bit Selects 1 and 0)

Together with the Sync Enable (CSR 1/Bit 3), these two bits allow the user to establish byte boundaries for the data stream; this is done in the following way. After bit synchronization is established, and the preamble pattern is verified, the 8X330 looks for a change in the normal preamble pattern. As shown in the following Truth Table, Bit Select 1 (Bit 2) and Bit Select 0 (Bit 3) identifies the bit cell within the first nibble of the first Address-Mark byte in which the first deviation from the normal preamble is expected. BYTRA is always referenced to bit cell 0.

BS 0	BS 1	BIT CELL
0	0	0
0	1	1
1	0	2
1	1	3

Bit 4 (Preamble Select)

This bit is used only for bit synchronization — refer to CSR 1/Bit 3. With Bit 1 of CSR 2 set to 0 (Read Mode) and the Preamble Select bit set to 0, the preamble field is assumed to be all zeroes; with the Preamble Select it set to 1, the preamble field is assumed to be all ones. In either case, preamble validity is determined by the 8X300.

Bits 5,6 (E1 and E2)

Together, E1 and E2 select the encoding scheme used to write data on the disk — refer to truth table that follows.

E1	E2	ENCODING SCHEME
1	X	FM
0	0	MFM
0	1	M^{2}FM

x = don't care

Bit 7 (1/2F)

This bit allows the data transfer rate to be changed without modification of the frequency-selective components in the data-separation logic; thus, differences in data transfer rates between standard-and-mini floppies can be accommodated via software — no component or other hardware changes. Assuming an 8MHz crystal and with the 1/2F bit set to 1, the data transfer rate is 250K bits per second in the single-density (FM) mode and 500K bits per second in the double-density (MFM/M^{2}FM) mode. When set to 0, the transfer rates are halved — 125K bits and 250K bits, respectively. When using frequencies other than 8MHz, the data transfer rate is determined as follows:

BIT 7 (½F)	SINGLE-DENSITY (FM)	DOUBLE-DENSITY (MFM/M^{2}FM)
0	$\frac{\text{xtal freq}}{64}$	$\frac{\text{xtal freq}}{32}$
1	$\frac{\text{xtal freq}}{32}$	$\frac{\text{xtal freq}}{16}$

Command/Status Register #3 (CSR 3/Address 134_8)

This register contains seven bits (Bit 0 through Bit 6) which determines the state of the disk-command outputs; writing to Bit 7 has no effect and reading Bit 7 always returns a zero. When a logical "1" is specified by the 8X300 program for a given disk-command line, a high will appear at the output of the 8X330 for that particular command line. Each bit and the output pin it controls are summarized below.

BIT (CSR 3)	CONTROL FUNCTION	PKG PIN NO
0	DC1 Output	12
1	DC2 Output	11
2	DC3 Output	10
3	DC4 Output	9
4	DC5 Output	8
5	DC6 Output	7
6	DC7 Output	6

Command/Status Register #4 (CSR 4/Address 135_8)

This register contains four bits (Bit 0 through Bit 3) which reflect the state of the disk-status inputs to the 8X330; reading all other bits (4 through 7) always returns a zero. These read-only bits and the reflected status they represent are as follows; the information specified by notation for Bit 7/CSR 1 is applicable to these input lines.

BIT (CSR 4)	CONTROL FUNCTION	PKG PIN NO
0	DS2 Input	16
1	DS3 Input	15
2	DS4 Input	14
3	DS5 Input	13

Sector Length Register — Address 136_8

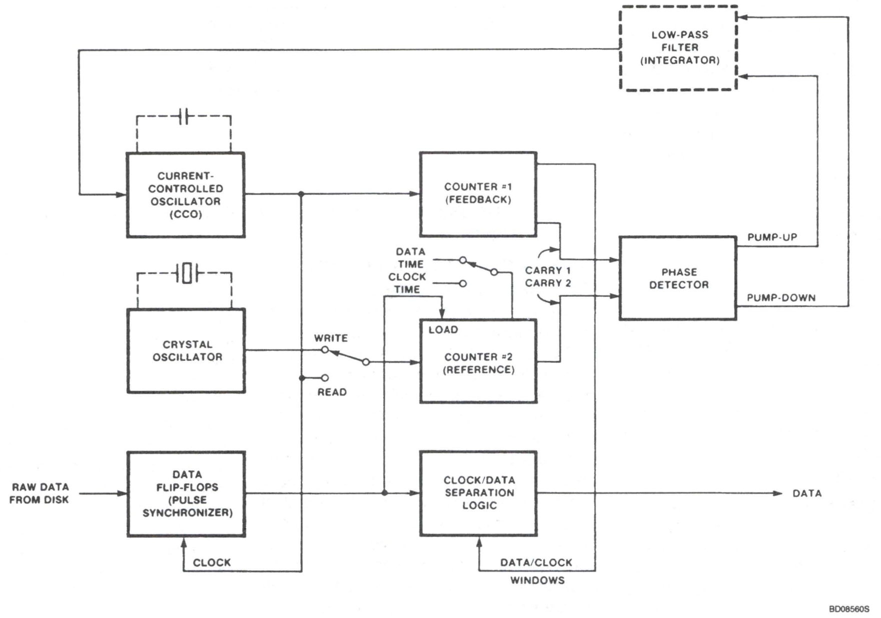
This register contains the load value for the lower eight (LSBs) bits of the Byte Counter. Data is transferred from the Sector Length Register to the Byte Counter under control of Load Counter Bit in CSR 1. When the contents of this register are transferred to another location via a read or write commands, the original holding of data is not lost; thus, if the same data is to be used more than once, a repetitive read or write can be implemented without reloading the register.

Data Register — Address 137_8

Together with the Data Shift Register, the Data Register is used for bidirectional transfer of data between the 8X330 and the I/O bus. All transfers to-and-from this register are made in conjunction with Bit 6 (BYTRA — Byte Transfer Flag) of CSR 1. When the Data Register Control bit (CSR 1/Bit 2) is set to 0, the content of this register is interleaved with four bits of data and four bits of clock. When data is transferred from the Data Register to the Data Shift Register, the original content of the Data Register is not lost.

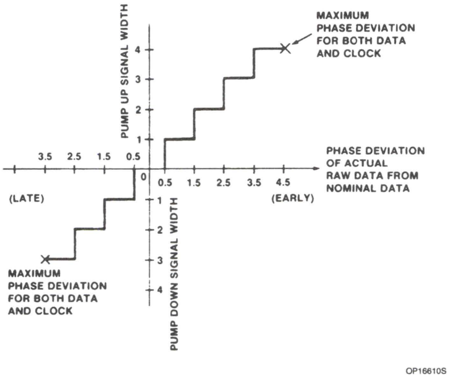
Phase Lock Loop (PLL) and Data Separation Logic

An expanded view of the phase-lock loop and the data-separation logic is shown in Figure 4. Basically, the PLL consists of two counters, a phase detector, and a feedback loop containing a low-pass filter (off-chip) that controls a phase-locked oscillator (CCO). In simplified form, the data-separation logic consists of data flip-flops (pulse synchronizer) and other circuits required to separate data and clock transitions. In the read mode, the output of the phase-locked oscillator (CCO) is applied to the clock inputs of counter #1, counter #2, and the pulse synchronization circuits. Essentially, the frequencies of the two counters are identical (phase relationships may or may not be identical); to maintain proper frequencies and to continuously correct any phase deviations, the following actions occur. Preset values which represent, respectively, nominal midpoints of the clock and data windows are present at counter #2 and, when an output appears at the pulse synchronizer, these preset values are entered. The count sequence for both counters is from "0 to F"; hence, the phase difference between Carry 1 (counter #1) and Carry 2 (counter #2) actually corresponds to any phase deviation between the CCO and the synchronized data from the disk. The phase detector measures the phase difference between the two carry inputs and produces a series of quantized pulses whose widths are proportional to the phase error at the end of each counting cycle. After integration by the low-pass filter, a current proportional to the phase error is applied to the current-controlled oscillator. Accordingly, the CCO is driven in a direction (pump-up or pump-down) to correct any phase difference between the synchronized disk data and the feedback-controlled clock. Phase detector characteristics for both single-and-double density formats are shown in Figures 5 and 6.


LOW-PASS FILTER (INTEGRATOR)
CURRENT-CONTROLLED OSCILLATOR (CCO)
COUNTER #1 (FEEDBACK)
PUMP-UP
DATA TIME
CLOCK TIME
CARRY 1
CARRY 2
PHASE DETECTOR
PUMP-DOWN
CRYSTAL OSCILLATOR
WRITE
READ
LOAD
COUNTER #2 (REFERENCE)
RAW DATA FROM DISK
DATA FLIP-FLOPS (PULSE SYNCHRONIZER)
CLOCK/DATA SEPARATION LOGIC
DATA
CLOCK
DATA/CLOCK WINDOWS
BD08560S

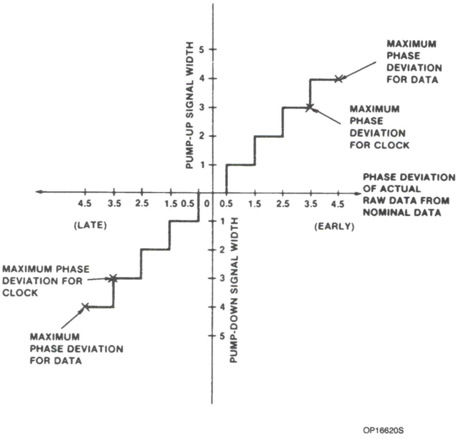
NOTE:
Components shown with dashed lines are located off-chip

Figure 4. Simplified Block of Phase-Lock Loop



NOTE:
All horizontal/vertical units are specified in CCO clock periods; for Standard Floppy FM Format, one CCO clock period equals 250 nanoseconds

Figure 5. Phase Detector Characteristic for Single-Density (FM) Format



NOTE:
All horizontal/vertical units are specified in CCO clock periods; for standard floppy double density (MFM/M^{2}FM), one CCO clock period equals 125 nanoseconds

Figure 6. Phase Detector Characteristic for Double-Density (MFM/M^{2}FM) Format

Data Processing and Error-Check Functions

The functions of the 8X330 are summarized in Figures 7 and 8. The read/write operations are software-controlled by previously-described bits of command/status registers CSR1 and CSR2. For the sake of simplicity, control lines and much of the control logic associated with the data processing and error-check functions are omitted in the read/write diagrams.

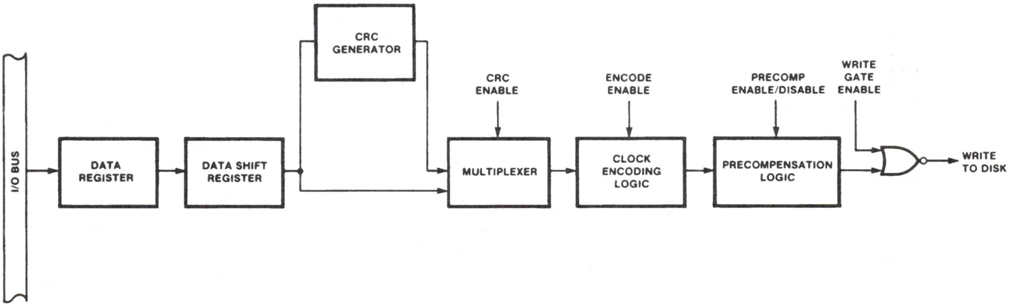

I/O BUS
DATA REGISTER
DATA SHIFT REGISTER
CRC GENERATOR
CRC ENABLE
MULTIPLEXER
ENCODE ENABLE
CLOCK ENCODING LOGIC
PRECOMP ENABLE/DISABLE
PRECOMPENSATION LOGIC
WRITE GATE ENABLE
WRITE TO DISK
BD08570S

Figure 7. Simplified Block of Data Processing and Error Check Functions — Write Mode

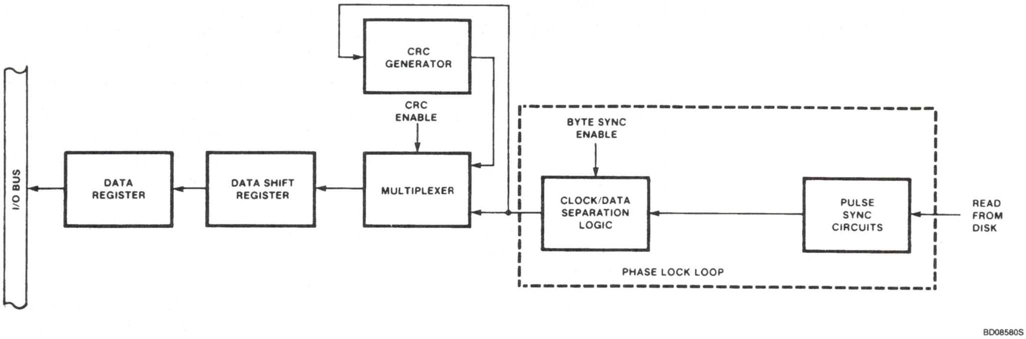

I/O BUS
DATA REGISTER
DATA SHIFT REGISTER
CRC GENERATOR
CRC ENABLE
MULTIPLEXER
BYTE SYNC ENABLE
CLOCK/DATA SEPARATION LOGIC
PULSE SYNC CIRCUITS
PHASE LOCK LOOP
READ FROM DISK
BD08580S

Figure 8. Simplified Block of Data Processing and Error Check Functions — Read Mode

DC ELECTRICAL CHARACTERISTICS V_{CC} = 5V (±5%), T_A = 0°C to 70°C

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNITS	COMMENTS
			Min	Typ	Max		
V_{IH}	High-level input voltage		2		V_{CC}	V	For all inputs except X1, X2, C1, C2, CCO, and V_{CR}
V_{IL}	Low-level input voltage		−1		0.8	V	
V_{CC}	Supply voltage		4.75	5	5.25	V	5V (±5%)
V_{CR}	Regulator voltage	$V_{CC} = 5V$		3.1		V	From series-pass transistor
V_{CL}	Input clamp voltage	V_{CC} = Min $I_{IN} = -5mA$	−1			V	Inputs X1, X2, C1, C2, and CCO do not have internal clamp diodes.
V_{OH}	High-level output voltage	V_{CC} = Min; $I_{OH} = -0.4mA$	2.7			V	DC1 through DC7 (Pins 6 – 12) & $\overline{DW}$ (Pin 4)
		V_{CC} = Min; $I_{OH} = -3mA$	2.4			V	$\overline{IV0}$ – $\overline{IV7}$ (Pins 25 – 32)
V_{OL}	Low-level output voltage	V_{CC} = Min; $I_{OL} = 8mA$			0.5	V	DC1 through DC7 (Pins 6 – 12); $\overline{PUP}$, $\overline{PDN}$ (Pins 33, 34); $\overline{DW}$ (Pin 4)
		V_{CC} = Min; $I_{OL} = 16mA$			0.55	V	$\overline{IV0}$ – $\overline{IV7}$ (Pins 25 – 32)
		V_{CC} = Min; $I_{OL} = 40mA$			0.55	V	$\overline{WG}$ (Pin 19)
I_{CEX}	Open-collector leakage current with output set to 1.	V_{CC} = Min; $V_{OUT} = V_{CC}$			100	μA	$\overline{WG}$ (Pin 19); $\overline{PUP}$ (Pin 34); $\overline{PDN}$ (Pin 33)
I_{IH}	High-level input current	V_{CC} = Max; $V_{IN} = 2.7V$			20	μA	DS1 – DS5 (Pins 13 – 17); $\overline{PF}$ (Pin 18); $\overline{DR}$ (Pin 5)
					40	μA	$\overline{ME}$ (Pin 21); MCLK (Pin 22); SC (Pin 23); WC (Pin 24)
		V_{CC} = Max; $V_{IN} = 5.25V$; CCO (Pin 35) input current = 0mA			4	mA	With C1 (Pin 36) under test, C2 (Pin 37) is open and, vice-versa.
		V_{CC} = Max; $V_{IN} = 5.25V$ CCO (Pin 35) input current = 1mA			2	mA	
		V_{CC} = Max; $V_{IN} = 0.6V$			4	mA	With X2 (Pin 2) under test, X1 (Pin 3) is open and, vice-versa.
		V_{CC} = Max; $V_{IN} = 4.5V$			50	μA	$\overline{IV0}$ – $\overline{IV7}$ (pins 25 – 32)
V_{CCO}	Input voltage for current-controlled oscillator	$V_{CC} = 5V$; $T_A = 25°C$ CCO input current (Pin 35) = 300μA		750		mV	
I_{IL}	Low-level input current	V_{CC} = Max; $V_{IN} = 0.4V$			−400	μA	DS1 – DS5 (Pins 13 – 17); $\overline{PF}$ (Pin 18); $\overline{DR}$ (Pin 5)
					−800	μA	$\overline{ME}$ (Pin 21); MCLK (Pin 22); SC (Pin 23); WC (Pin 24)
					−4	mA	X1 (Pin 2), X2 (Pin 3), with X1 under test, X2 is open and, vice-versa.
		V_{CC} = Max $V_{IN} = 0.5V$			−550	μA	$\overline{IV0}$ – $\overline{IV7}$ (Pins 25 – 32)

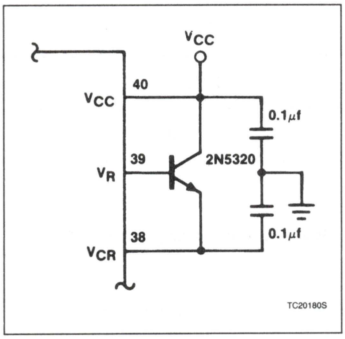
Floppy Disk Formatter/Controller 8X330

DC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNITS	COMMENTS
			Min	Typ	Max		
I_{OS}	Output short-circuit current	V_{CC} = Max; Output = "1"; V_{OUT} = "0".	−15		−100	mA	DC1 − DC7 (Pins 6 − 12) & $\overline{DW}$ (Pin 4)
		(NOTE: At any time, no more than one output should be connected to ground)	−30		−140	mA	$\overline{IV0}$ − $\overline{IV7}$ (Pins 25 − 32)
I_{CC}	(Pin 40)	V_{CC} = Max			200	mA	
I_{CR}		V_{CC} = Max			250	mA	
I_{REG}	(Pin 39)	V_{CC} = 5V; V_{CR} = 0V & V_R = 2V	−16		−27	mA	

NOTES:

1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. All voltages measured with respect to ground terminal.
3. Unless otherwise specified, each test requires that V_{CR} be supplied through a series-pass transistor as shown in the accompanying drawing.



AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V(\pm 5\%)$, $T_A = 0°C$ to $70°C$

MNEMONIC	REFERENCE	INPUT	OUTPUT	TYPICAL MINIMUM	TYPICAL MAXIMUM	COMMENTS
t_{PD}		↓ $\overline{PF}$	$\overline{WG}$ ↑		60ns	Refer to Note 3 and Test Loading Circuit #1.
t_{PD}		↑ MCLK	$\overline{WG}$ ↑		100ns	
t_{PD}		↑ MCLK	$\overline{WG}$ ↓		100ns	
t_{PD}		↑ MCLK	DC1 – 7 ↑		70ns	Refer to Note 3 and Test Loading Circuit #2.
t_{PD}		↑ MCLK	DC1 – 7 ↓		70ns	
t_{pw}		↓ $\overline{PF}$ ↑		50ns		
t_{pw}		↓ $\overline{DR}$ ↑		50ns		
t_{pw}			↓ $\overline{DW}$ ↑			Note 1
t_{SETUP}	↓ SC	Input on DS1 – 5		55ns		Note 2
t_{SETUP}	↓ WC	Input on DS1 – 5		55ns		Note 2
t_{SETUP}	↓ $\overline{ME}$	Input on DS1 – 5		55ns		Note 2
t_{HOLD}	↓ SC	Input on DS1 – 5		0ns		Note 2
t_{HOLD}	↓ WC	Input on DS1 – 5		0ns		Note 2
t_{HOLD}	↓ $\overline{ME}$	Input on DS1 – 5		0ns		Note 2
t_{OE} — $\overline{ME}$, SC & WC		↓ $\overline{ME}$ ↓ SC ↓ WC	I/O bus		25ns	Refer to Test Loading Circuit #3.
t_{OD} — $\overline{ME}$, SC & WC		$\overline{ME}$ ↑ SC ↑ WC ↑	I/O bus (3-State)		30ns	
t_w (MCLK pulse width)		↑ MCLK ↓		45ns		
t_{SD} (data setup time)	MCLK ↓	I/O bus		50ns		
t_{SD} ($\overline{ME}$ setup time)	MCLK ↓	$\overline{ME}$		45ns		
t_{SD} (SC setup time)	MCLK ↓	SC		45ns		
t_{SD} (WC setup time)	MCLK ↓	WC		45ns		
t_{HD} (data hold time)	MCLK ↓	I/O bus		0ns		
t_{HD} ($\overline{ME}$ hold time)	MCLK ↓	$\overline{ME}$		0ns		
t_{HD} (SC hold time)	MCLK ↓	SC		0ns		
t_{HD} (WC hold time)	MCLK ↓	WC		0ns		

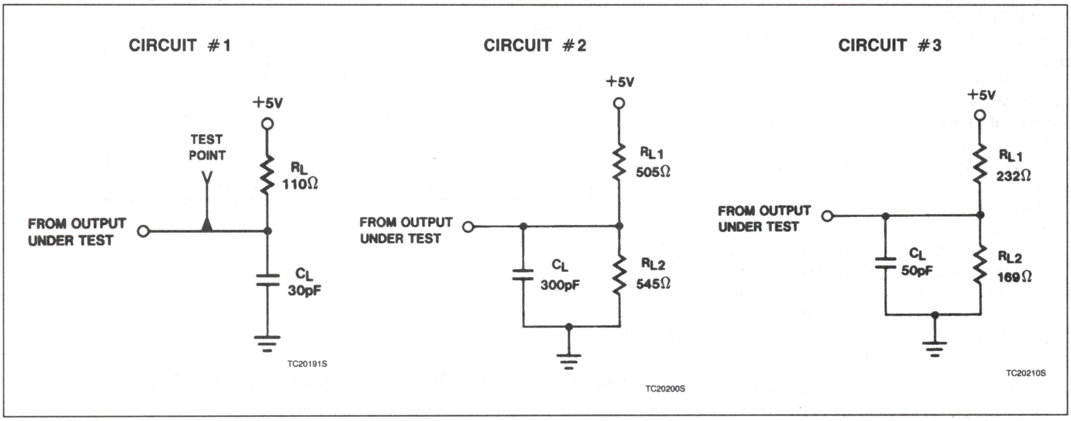
NOTES:

The values of these parameters were determined by device characterization procedures. They are, however, neither tested nor guaranteed.

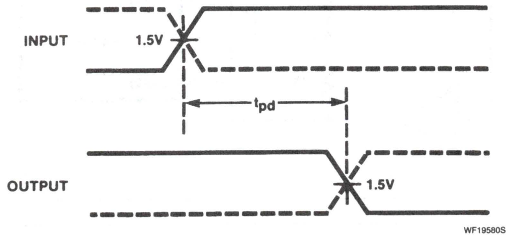
1. Write pulse width = $2/F_{XTAL}$, that is, or 8MHz crystal, $t_{pw} = 250\eta sec$ (typical)
2. Changes on DS1 – 5 are not stored in read mode ($\overline{ME} = 0$, SC = 0, and WC = 0)
3. During the period when MCLK is high, measurement is made with $\overline{ME}$ = Low, SC = Low, and WC = High.

Floppy Disk Formatter/Controller **8X330**

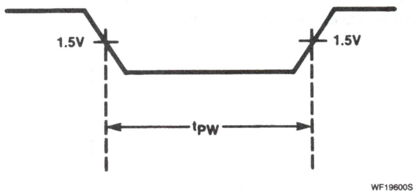
TEST LOADING CIRCUIT


CIRCUIT #1
+5V
TEST POINT
R_L 110Ω
FROM OUTPUT UNDER TEST
C_L 30pF
TC20191S
CIRCUIT #2
+5V
R_{L1} 505Ω
FROM OUTPUT UNDER TEST
C_L 300pF
R_{L2} 545Ω
TC20200S
CIRCUIT #3
+5V
R_{L1} 232Ω
FROM OUTPUT UNDER TEST
C_L 50pF
R_{L2} 169Ω
TC20210S

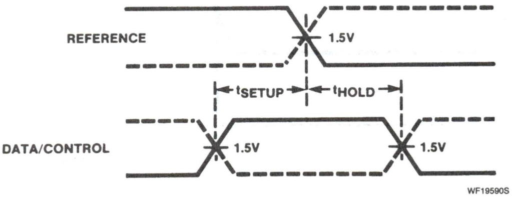
PROPAGATION DELAY TIME


INPUT
1.5V
t_{pd}
OUTPUT
1.5V
WF19580S

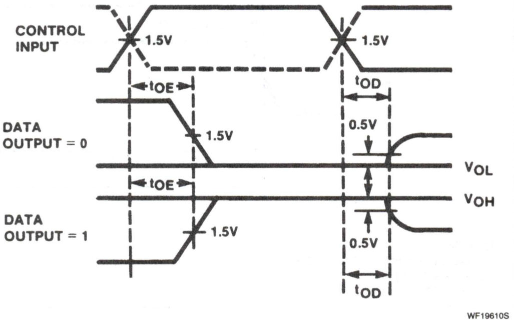
PULSE WIDTH WAVEFORM


1.5V
1.5V
t_{PW}
WF19600S

SETUP & HOLD TIMES


REFERENCE
1.5V
t_{SETUP}
t_{HOLD}
DATA/CONTROL
1.5V
1.5V
WF19590S

OUTPUT ENABLE AND DISABLE TIMES $\overline{IV0}$-$\overline{IV7}$


CONTROL INPUT
1.5V
1.5V
t_{OE}
t_{OD}
DATA OUTPUT = 0
1.5V
0.5V
V_{OL}
t_{OE}
V_{OH}
DATA OUTPUT = 1
1.5V
0.5V
t_{OD}
WF19610S

CLOCK REQUIREMENTS

Crystal Oscillator

The on-chip crystal oscillator circuit is designed for operation using an external series-resonant quartz crystal; alternately the crystal oscillator can be driven with complementary outputs of a pulse generator or interfaced to a master clock source via TTL logic — see accompanying circuits. When a crystal is used, the on-chip oscillator operates at the resonant frequency (f.) of the crystal; the crystal connects to the 8X330 via pins 3(X1) and 2 (X2). The lead lengths of the crystal should be approximately equal and as short as possible; also, avoid close proximity to all potential noise sources. The crystal should be hermetically sealed (HC type can) and have the following electrical characteristics:

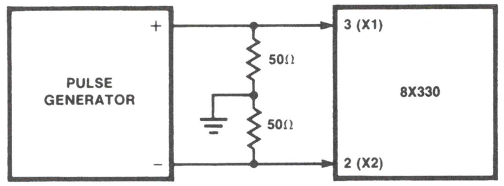
Type: Fundamental mode, series resonant

Impedance at Fundamental:
35-Ω maximum

Impedance at Harmonics and Spurs:
50-Ω minimum

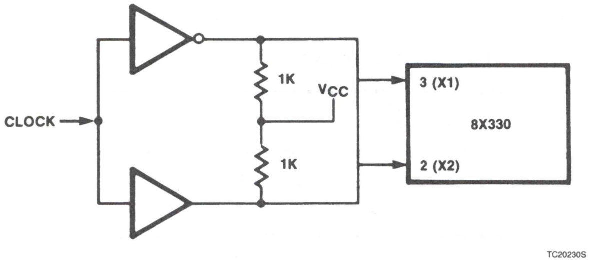
When the crystal oscillator is externally-driven, typical waveforms are as follows:

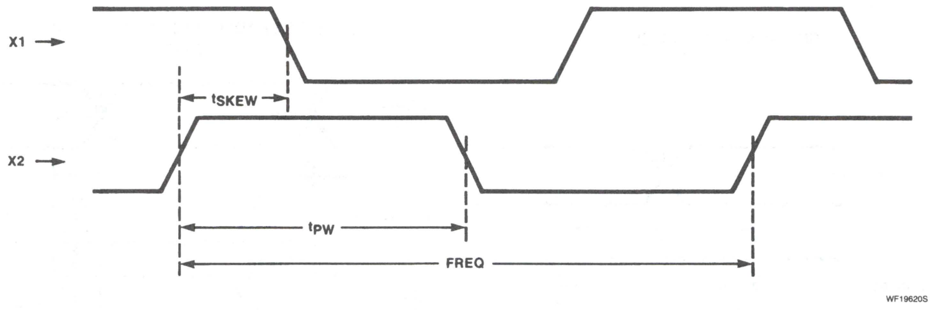
CLOCKING XTAL OSC WITH PULSE GEN



NOTES:
PULSE GEN CHAR:
$Z_{OUT} = 50\Omega$
$V_{OUT} = 0 - 2V$
RISETIME ≤ 10ns
SKEW ≤ 10ns
COMPLEMENTARY OUTPUTS

CLOCKING XTAL OSC WITH OPEN-COLLECTOR TTL



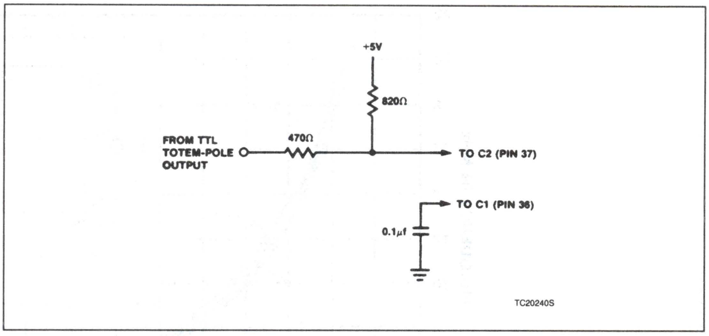


NOTES:
FREQUENCY = 8MHz (TYP) 16MHz (MAX)
t_{PW} = 50% (TYP)
$t_{SKEW} \leq$ 10ns

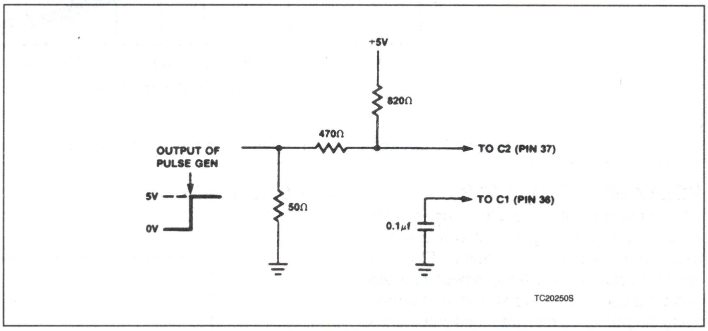
Current-Controlled Oscillator (CCO).

A non-polarized ceramic or mica capacitor is recommended for the current-controlled oscillator. The capacitor connects to the 8X330 via pins 37 (C2) and 36 (C1); lead lengths of the capacitor should be approximately the same and as short as possible. When the input current to the CCO is near zero (maximum frequency), the capacitor value should be chosen so that the high-limit rest frequency of the oscillator does not exceed 24MHz. If the rest frequency is higher than 24MHz, synchronization of the CCO with the crystal oscillator just prior to the read operation, may be impeded. The curves in Figure 9 (current-versus-frequency) and Figure 8 (capacitance-versus-frequency) show how these design parameters affect operation of the CCO over a temperature range of 0°C to 70°C. A suitable test circuit for verification/validation of the current-controlled oscillator is also shown in Figure 10. Like the crystal oscillator, the CCO can be driven with the TTL output of a pulse generator or interfaced to a master clock via TTL logic — see accompanying diagrams.

CLOCKING WITH OPEN-COLLECTOR TTL


+5V
820Ω
FROM TTL TOTEM-POLE OUTPUT
470Ω
TO C2 (PIN 37)
TO C1 (PIN 36)
0.1µf
TC20240S

CLOCKING WITH PULSE GENERATOR


+5V
820Ω
470Ω
OUTPUT OF PULSE GEN
TO C2 (PIN 37)
5V
0V
50Ω
TO C1 (PIN 36)
0.1µf
TC20250S

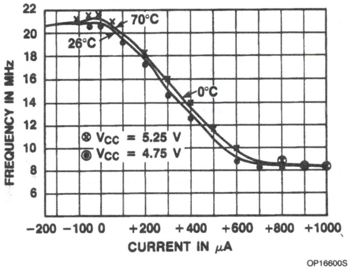
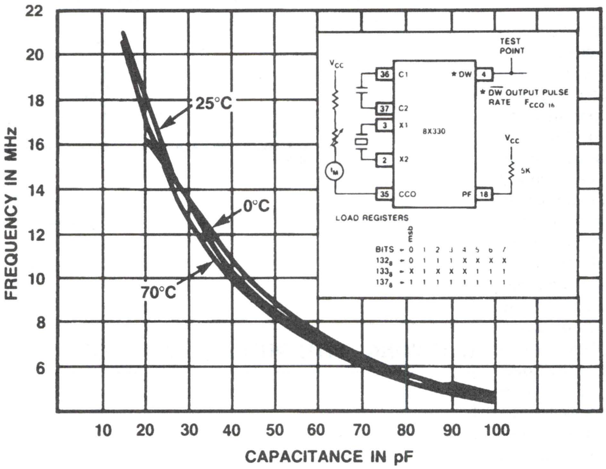


Figure 9. Current-versus-Frequency With: $V_{CC} = 5V$ and Capacitance = 25pF



OP16590S

Figure 10. Capacitance-versus-Frequency With: $V_{CC} = 5V$, $VCR = 2.5V$, and $1 = 300\mu A$

VOLTAGE REGULATOR

All internal logic of the 8X330 is powered by an on-chip voltage regulator that requires an external series-pass transistor. Electrical specifications for the off-chip power transistor and a typical hook-up are shown in accompanying diagrams. To minimize lead inductance, the transistor should be as close as possible to the 8X330 package and the emitter should be ac-grounded via a 0.1-microfarad capacitor.

ORDERING INFORMATION

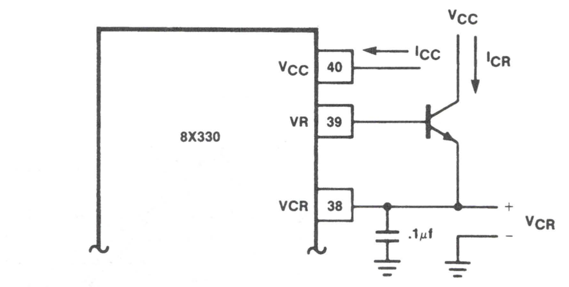
Order number: N8X330

Package information: Refer to Signetics price list

Supply voltage: 5V (±5%)

Operating temperature range: 0°C to +70°C

TYPICAL HOOK-UP



TC20260S

ELECTRICAL SPECIFICATIONS

*PARAMETER	CONDITIONS	LIMITS
h_{fe}	$V_{CE} = 2V$	> 50
V_{BEON}	$V_{CE} = 5V/I_C = 500mA$	< 1V
V_{CESAT}	$I_C = 500mA/I_B = 50mA$	< 0.5V
BV_{CEO}		> 8V
f_t		> 30MHz

NOTE:
*Medium power NPN silicon ($0° < T_A < 70°C$) recommended parts: 2N5320, 2N5337.

Signetics

8X353
Bipolar Ram (32 × 8)

Product Specification

Microprocessor Products

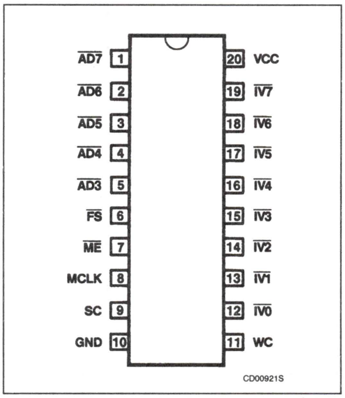
DESCRIPTION

The 8X353 is a 32-byte RAM designed principally as a working storage element in 8X305-based systems. The 8X353 is ideal for applications requiring a relatively small amount of data storage and maximum I/O flexibility. Since the 8X353 takes up only 32 of 256 locations on a controller bank, this device allows single cycle, bank-to-bank data transfer and other implementations in which the user does not wish to dedicate an entire I/O bank to data storage. Contributing to the versatility of the 8X353 is a fast select feature which, when supplemented with extended micro-code, allows this device to be selected externally of the IV bus.

FEATURES

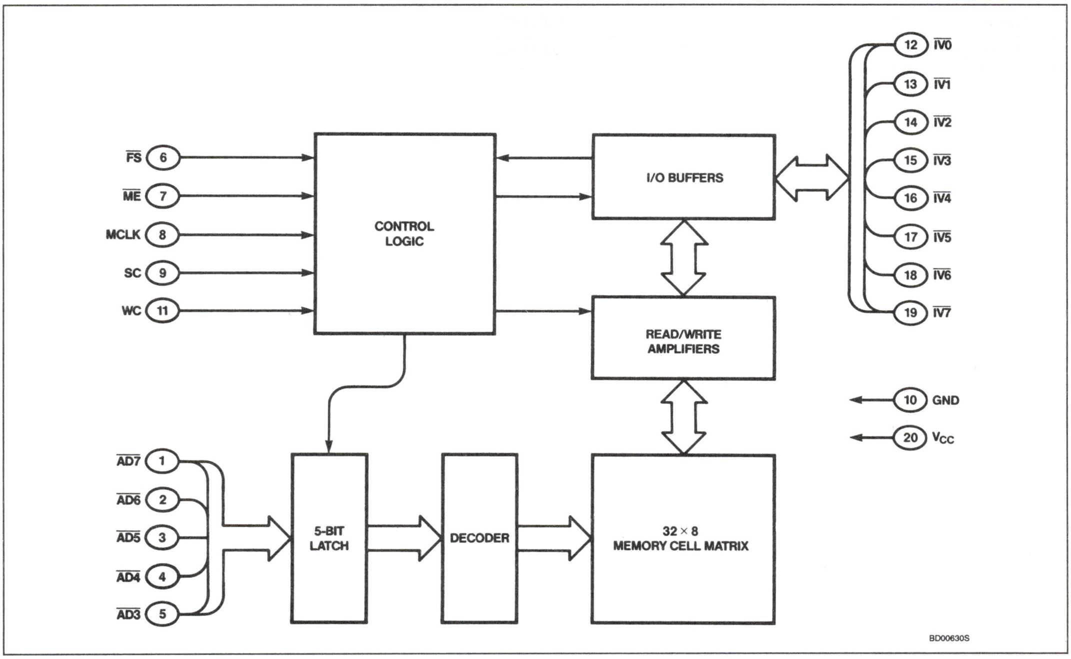
- **32 bytes of storage**
- **32 port addresses $(00_8 - 37_8)$**
- **224 bank addresses available for other 8X305 peripherals**
- **Fast Select feature for use with extended micro-code**
- **On-chip address decoding**
- **Separate address input pins**
- **Direct interface**
- **Single 5V power supply**
- **0.3 inch, Slim Line package**
- **0.3 inch, CERDIP package**

PIN CONFIGURATION



AD7 1
AD6 2
AD5 3
AD4 4
AD3 5
FS 6
ME 7
MCLK 8
SC 9
GND 10
20 VCC
19 IV7
18 IV6
17 IV5
16 IV4
15 IV3
14 IV2
13 IV1
12 IV0
11 WC
CD00921S

BLOCK DIAGRAM



FS 6
ME 7
MCLK 8
SC 9
WC 11
CONTROL LOGIC
I/O BUFFERS
READ/WRITE AMPLIFIERS
12 IV0
13 IV1
14 IV2
15 IV3
16 IV4
17 IV5
18 IV6
19 IV7
10 GND
20 VCC
AD7 1
AD6 2
AD5 3
AD4 4
AD3 5
5-BIT LATCH
DECODER
32 × 8 MEMORY CELL MATRIX
BD00630S

Bipolar Ram (32 × 8) 8X353

PIN DESCRIPTION

PIN NO.	IDENTIFIER	FUNCTION
1 – 5	$\overline{AD7}$ – $\overline{AD3}$	**ADdress bus** – Active Low inputs that define which memory locations are to be accessed.
6	$\overline{FS}$	**Fast Select** – When active Low, this input allows read and write operations without performing a full select cycle.
7	$\overline{ME}$	**Master Enable** – When Low, this input indicates the microcontroller bank is active; otherwise the chip will not respond to any signal on the IV bus.
8	MCLK	**Master CLocK** – A standard 8X305 clock input used for timing reference and system synchronization.
9	SC	**Select Command** – When active High, this input indicates IV bus data is to be interpreted as an address.
10	GND	**GrouND.**
11	WC	**Write Command** – When active High, this signal indicates data is being input from the IV bus.
12 – 19	$\overline{IV0}$ – $\overline{IV7}$	**Interface Vector Bus** – 3-State, active Low, bidirectional signals which communicate I/O data and addresses (chip select).
20	V_{CC}	**Supply Voltage.**

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-pin plastic DIP	N8X353N
20-pin CERDIP	N8X353F

Bipolar Ram (32 × 8) 8X353

FUNCTIONAL OPERATION

The 8X353 is capable of performing two functions — write and read. These operations, and the control logic necessary for their execution, are described in the following text and summarized in Table 1. Typical timing relationships for write, read, and select cycles are shown in Timing Diagrams.

Select Logic

Standard Select Cycle — before a Read or Write can occur, the 8X353 must first be selected by either a standard select cycle or by Fast Select operation (see below). A standard select cycle is initiated when SC and MCLK are high, and $\overline{ME}$ and WC are low. The three most significant bits on the $\overline{IV}$ bus are then compared with fixed internal values (110_2 active low), and if identical an internal select latch is activated (high) and the five bits of the memory address (pins AD7 – AD3) are loaded into an address latch.

As shown in Table 1, the 8X353 remains in the selected state until another select cycle is initiated, at which point the select latch and addresses are updated.

In situations where more than one RAM is required, the fixed internal values of IV0, IV1, and IV2 can be used as chip addresses. By scrambling these pins, up to three RAMs can be selected from a single IV bus.

Write/read Operation

Write and read operations are possible only after the 8X353 has been selected by either a standard select cycle or by Fast Select operation. As shown in Table 1, data I/O is controlled by $\overline{ME}$, SC, WC, MCLK, FS, and the current state of the select latch.

After an address has been latched, data is written to the specified memory location when MCLK and WC are high and $\overline{ME}$ is low; a read occurs when MCLK, WC, SC, and $\overline{ME}$ are low. In both cases, data is transmitted via the IV bus.

Table 1. Summary of 8X353 Operation

MODE	CONDITIONS						RESULTS				
	$\overline{ME}$	SC	WC	MCLK	$\overline{FS}$	Select Latch	$\overline{IV}$ Bus	$\overline{AD}$ Bus	Data	Address Latch	Select Latch
A_{US}	L	H	L	H	X	X	Input (chip address)	Input[2]	Keep	Update[2]	Update[2]
D_{IS}	L	L	H	H	H	H	Input	Ignore	Update	Keep	Keep
D_{OS}	L	L	L	L	H	H	Output	Ignore	Keep	Keep	Keep
A_{UF}	L	S	L	H	L	X	Ignore	Input	Keep	Update	Keep
A_{UF}	H	X	X	H	L	X	Ignore	Input	Keep	Update	Keep
D_{IF}	L	L	H	H	L	X	Input	Ignore	Update	Keep	Keep
D_{OF}	L	L	L	L	L	X	Output	Ignore	Keep	Keep	Keep
$\overline{ME}$ High	H	X	X	X	H	X	Ignore	Ignore	Keep	Keep	Keep
$\overline{ME}$ High	H	X	X	L	L	X	Ignore	Ignore	Keep	Keep	Keep
D_{OS} w/o MCLK Low	L	L	L	H	H	X	Ignore	Ignore	Keep	Keep	Keep
D_{OS} w/o select latch High	L	L	L	L	H	L	Ignore	Ignore	Keep	Keep	Keep
D_{IS} w/o MCLK High	L	L	H	L	X	X	Ignore	Ignore	Keep	Keep	Keep
D_{IS} w/o select latch High	L	L	H	H	H	L	Ignore	Ignore	Keep	Keep	Keep
A_{US} w/o MCLK High	L	H	L	L	X	X	Ignore	Ignore	Keep	Keep	Keep
Not defined	X	H	H	X	X	X	Not defined				

NOTES:

1. A_{US} = address update with standard select
   D_{IS} = data input with standard select
   D_{OS} = data output with standard select
   A_{UF} = address update with fast select
   D_{IF} = data input with fast select
   D_{OF} = data output with fast select
2. Depending on IV bus data
3. Fast select Logic ($\overline{FS}$) given but is not used for 8X305 operation.
   X = Don't care

Bipolar Ram (32 × 8)

8X353

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_O	Off-state voltage	+5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to+150	°C

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, $0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	TEST CONDITIONS[1]	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				0.8	V
V_{OH}	High-level output voltage	IV pins: V_{CC} = Min, I_{OH} = −3.2mA	2.4			V
V_{OL}	Low-level output voltage	IV pins : V_{CC} = Min, I_{OL} = 16mA			0.55	V
V_{IC}	Input clamp voltage	V_{CC} = Min, I_{IN} = −10mA			−1.5	V
I_{IH}	High-level input (or leakage) current	Address pins and control pins: V_{CC} = Max, V_{IN} = 4.5V			100	μA
I_{IL}	Low-level input current	Address pins and control pins: V_{CC} = Max, V_{IN} = 0.5V			−100	μA
I_{OS}	Short circuit output current	IV pins: V_{CC} = Max[2]	−30		−140	mA
I_{OZH}	High-Z state output current – high level	IV pins: V_{CC} = Max, V_0 = 2.7V			100	μA
I_{OZL}	High-Z state output current – low level	IV pins: V_{CC} = Max, V_0 = .5V			−200	μA
I_{CC}	Supply current	All inputs: V_{CC} = Max, V_{IN} = 0V			200	mA

NOTES:

1. All voltages measured with respect to Ground terminal.
2. At any time, no more than one output should be connected to ground.

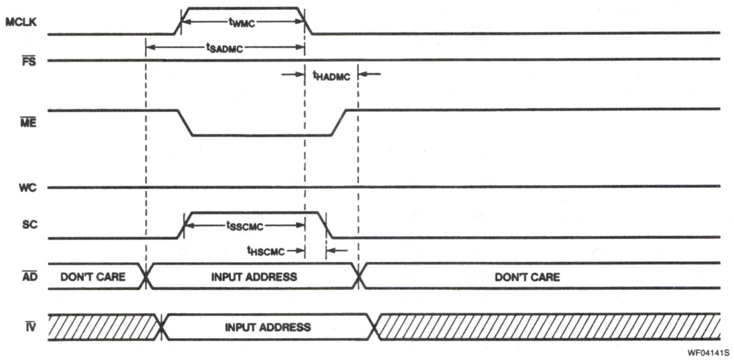
AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%$, $0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS[1, 2, 3]	LIMITS			UNIT
		From	To		Min	Typ	Max	
Pulse width								
t_{WMC}	MCLK pulse width	↑MCLK	↓MCLK	FS = High; $\overline{ME}$ = Low	35			ns
Set-up times								
t_{SIVMC}	IV setup time	$\overline{IV}$	↓MCLK	FS = High; $\overline{ME}$ = Low	40			ns
t_{SADMC}	Address setup time	$\overline{AD}$	↓MCLK	FS, SC = High; WC,$\overline{ME}$ = Low	40			ns
t_{SWCMC}	Write command setup time	↑WC	↓MCLK	FS = High; SC,$\overline{ME}$ = Low	60			ns
t_{SSCMC}	Select command setup time	↑SC	↓MCLK	FS = High; WC,$\overline{ME}$ = Low	55			ns
t_{SMEMC}	Master enable setup time	↓$\overline{ME}$	↓MCLK	FS = High	55			ns
Hold times								
t_{HIVMC}	IV hold time	↓MCLK	$\overline{IV}$	FS = High; $\overline{ME}$ = Low	5			ns
t_{HADMC}	Address hold time	↓MCLK	$\overline{AD}$	FS, SC = High; WC,$\overline{ME}$ = Low	5			ns
t_{HWCMC}	Write command hold time	↓MCLK	↓WC	FS = High; SC,$\overline{ME}$ = Low	2			ns
t_{HSCMC}	Select command hold time	↓MCLK	↓SC	FS = High; WC,$\overline{ME}$ = Low	0			ns
t_{HMEMC}	Master enable hold time	↓MCLK	↑$\overline{ME}$	FS = High; SC, WC = Low	5			ns
Output enable times								
t_{EDOMC}	IV output enable time	↓MCLK	$\overline{IV}$	FS = High; WC,$\overline{ME}$, SC = Low			30	ns
t_{EDOWC}	IV output enable time	↓WC	$\overline{IV}$	FS = High; MCLK,$\overline{ME}$, SC = Low			30	ns
t_{EDOSC}	IV output enable time	↓SC	$\overline{IV}$	FS = High; MCLK, WC,$\overline{ME}$ = Low			30	ns
t_{EDOME}	IV output enable time	↓$\overline{ME}$	$\overline{IV}$	FS = High; MCLK, WC, SC = Low			30	ns
Output disable times								
t_{DDOMC}	IV output disable time	↑MCLK	$\overline{IV}$	FS = High; WC,$\overline{ME}$, SC = Low[4]			25	ns
t_{DDOWC}	IV output disable time	↑WC	$\overline{IV}$	FS = High; MCLK,$\overline{ME}$ = Low[4]			25	ns
t_{DDOSC}	IV output disable time	↑SC	$\overline{IV}$	FS = High; MCLK, WC,$\overline{ME}$ = Low[4]			25	ns
t_{DDOME}	IV output disable time	↑$\overline{ME}$	$\overline{IV}$	FS = High; MCLK, WC = Low[4]			25	ns

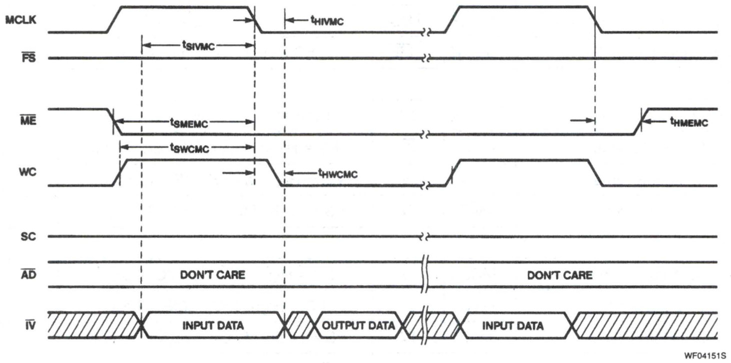
NOTES:

1. Loading: see Test Loading Circuits
2. All inputs are driven between 0 and 3V.
3. Except for output disable times, all timing parameters are measured at 1.5V.
4. These parameters are measured with a capacitive loading of 50pF and represent the output driver turn-off time. Disable times measured into a capacitive loading of 300pF will be a maximum of 40ns.

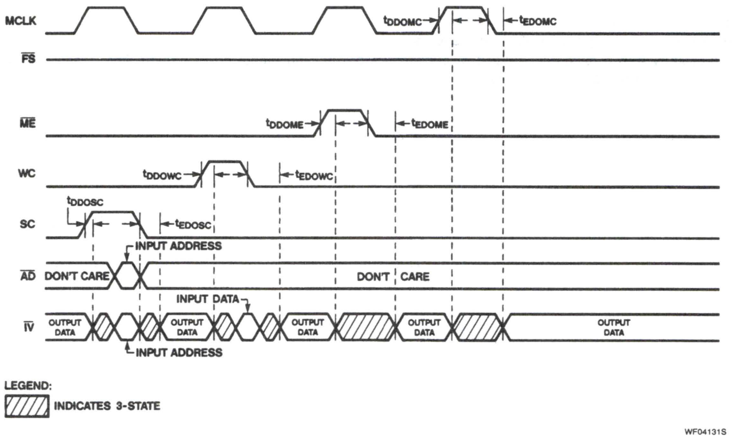
TIMING DIAGRAMS


MCLK
FS
ME
WC
SC
AD
IV
tWMC
tSADMC
tHADMC
tSSCMC
tHSCMC
DON'T CARE
INPUT ADDRESS
DON'T CARE
INPUT ADDRESS
WF04141S

a. Standard Select Cycle Timing

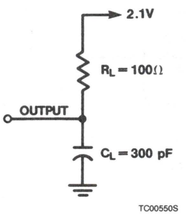

MCLK
FS
ME
WC
SC
AD
IV
tHIVMC
tSIVMC
tSMEMC
tHMEMC
tSWCMC
tHWCMC
DON'T CARE
DON'T CARE
INPUT DATA
OUTPUT DATA
INPUT DATA
WF04151S

b. Write/Read Cycle Timing


MCLK
FS
ME
WC
SC
AD
IV
tDDOMC
tEDOMC
tDDOME
tEDOME
tDDOWC
tEDOWC
tDDOSC
tEDOSC
INPUT ADDRESS
DON'T CARE
DON'T CARE
INPUT DATA
OUTPUT DATA
OUTPUT DATA
OUTPUT DATA
OUTPUT DATA
OUTPUT DATA
INPUT ADDRESS
LEGEND:
▨ INDICATES 3-STATE
WF04131S

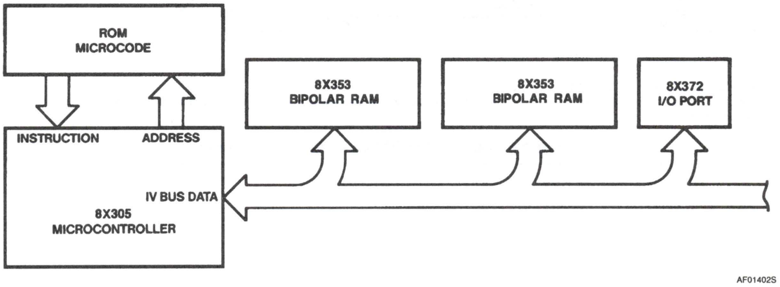
c. Output Enable/Disable Timing

TEST LOADING CIRCUIT

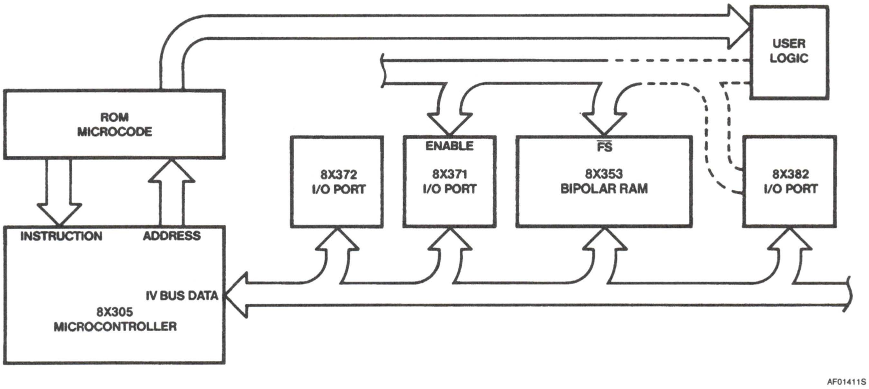

2.1V
$R_L = 100\Omega$
OUTPUT
$C_L = 300$ pF
TC00550S

NOTE:
C_L = 50pF for output disable times.

APPLICATIONS DIAGRAMS


ROM
MICROCODE
INSTRUCTION
ADDRESS
IV BUS DATA
8X305
MICROCONTROLLER
8X353
BIPOLAR RAM
8X353
BIPOLAR RAM
8X372
I/O PORT
AF01402S

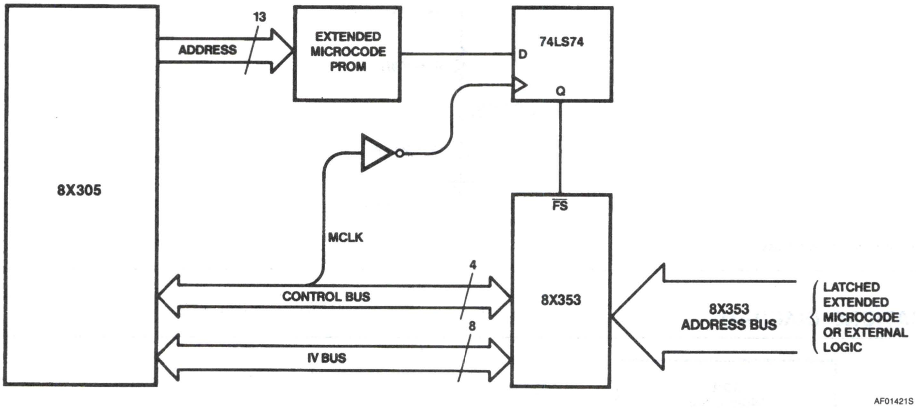
a. Implementation of the 8X353 Using $\overline{\text{IV}}$ Bus Select


USER
LOGIC
ROM
MICROCODE
INSTRUCTION
ADDRESS
IV BUS DATA
8X305
MICROCONTROLLER
8X372
I/O PORT
ENABLE
8X371
I/O PORT
$\overline{\text{FS}}$
8X353
BIPOLAR RAM
8X382
I/O PORT
AF01411S

b. Implementation of the 8X353 Using FAST Select and External Logic

Bipolar Ram (32 × 8) 8X353

APPLICATIONS DIAGRAM (Continued)


8X305
ADDRESS
13
EXTENDED MICROCODE PROM
74LS74
D
Q
MCLK
FS
4
CONTROL BUS
8
IV BUS
8X353
8X353 ADDRESS BUS
LATCHED EXTENDED MICROCODE OR EXTERNAL LOGIC
AF01421S

NOTES:
1. The control bus consists of MCLK, SC, WC and $\overline{\text{ME}}$.
2. The 8X353 can be selected using the $\overline{\text{FS}}$ pin instead of a normal select cycle. Since the flip-flop is clocked by the falling edge of MCLK, the data input to the flip-flop must have sufficient setup time.

Example Use of the 8X353 Fast Select Mode.

Signetics

8X355
LIFO Stack Memory (32 × 8)

Product Specification

Microprocessor Products

DESCRIPTION

The 8X355 Last In/First Out (LIFO) stack memory is designed for use with the 8X305 Microcontroller; however, it can be easily adapted to other microprocessor-based systems. The 32-byte storage capacity of the chip can be expanded by cascading to form 8-bit wide memories of any required depth. Typical applications are shown at the rear of this data sheet.

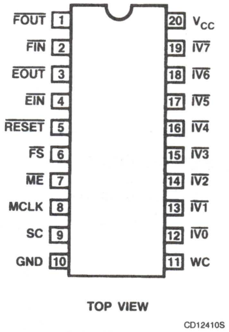
The 8X355 can be addressed by using standard $\overline{IV}$ bus control logic or, in high-performance systems, the instruction word of the Microcontroller can be extended and the 8X355 can be addressed in the fast-select mode. In the latter case, an address select cycle prior to each data access is omitted.

When used with the 8X310 interrupt Control Coprocessor, this enhanced throughput feature is very convenient for saving 8X305 internal registers during interrupt servicing. For either single or cascaded configurations, the 8X355 uses a fixed enabling address (130_8), thus, once enabled, the memory can accept an uninterrupted stream of data.

FEATURES

- **32 bytes of storage**
- **Cascadable LIFO operation**
- **Dedicated port address**
- **Fast select feature for use with extended microcode**
- **3-State TTL outputs**
- **Single 5-volt power supply**
- **0.3-inch, Slim Line package**

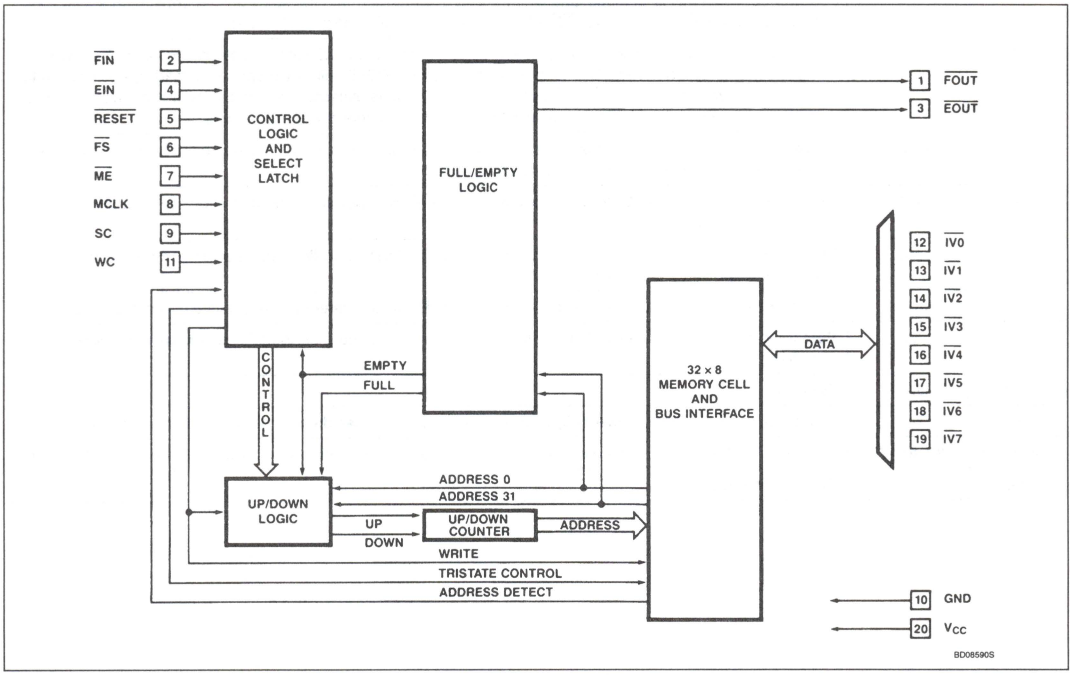
PIN CONFIGURATION



$\overline{FOUT}$ 1
$\overline{FIN}$ 2
$\overline{EOUT}$ 3
$\overline{EIN}$ 4
$\overline{RESET}$ 5
$\overline{FS}$ 6
$\overline{ME}$ 7
MCLK 8
SC 9
GND 10
20 V_{CC}
19 $\overline{IV7}$
18 $\overline{IV6}$
17 $\overline{IV5}$
16 $\overline{IV4}$
15 $\overline{IV3}$
14 $\overline{IV2}$
13 $\overline{IV1}$
12 $\overline{IV0}$
11 WC
TOP VIEW
CD12410S

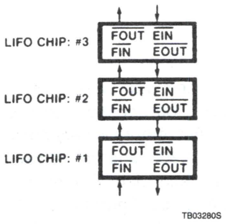
Order Numbers
N8X355N, N8X355F

BLOCK DIAGRAM



$\overline{FIN}$ 2
$\overline{EIN}$ 4
$\overline{RESET}$ 5
$\overline{FS}$ 6
$\overline{ME}$ 7
MCLK 8
SC 9
WC 11
CONTROL LOGIC AND SELECT LATCH
FULL/EMPTY LOGIC
1 $\overline{FOUT}$
3 $\overline{EOUT}$
CONTROL
EMPTY
FULL
UP/DOWN LOGIC
ADDRESS 0
ADDRESS 31
UP
DOWN
UP/DOWN COUNTER
ADDRESS
WRITE
TRISTATE CONTROL
ADDRESS DETECT
32 x 8 MEMORY CELL AND BUS INTERFACE
DATA
12 $\overline{IV0}$
13 $\overline{IV1}$
14 $\overline{IV2}$
15 $\overline{IV3}$
16 $\overline{IV4}$
17 $\overline{IV5}$
18 $\overline{IV6}$
19 $\overline{IV7}$
10 GND
20 V_{CC}
BD08590S

CASCADED CONFIGURATION



PIN DESCRIPTION

PIN NO	IDENTIFIER	FUNCTION
1	$\overline{\text{FOUT}}$	**Full Output:** When active (Low), this output indicates that the 8X355 is full and no further writes can be accepted until at least one read operation is executed. On power up, $\overline{\text{FOUT}}$ is High and remains in this inactive state until the last memory location is written to. In a cascaded configuration, $\overline{\text{FOUT}}$ is tied to $\overline{\text{FIN}}$ of the next higher LIFO chip in the stack.
2	$\overline{\text{FIN}}$	**Full Input:** In cascaded configurations, an active (Low) $\overline{\text{FIN}}$ signal indicates to a given LIFO chip that the next lower chip is full. When $\overline{\text{FIN}}$ is inactive, the 8X355 cannot be accessed except for select and deselect cycles. In non-cascaded configurations, $\overline{\text{FIN}}$ is usually forced low to enable reads and writes.
3	$\overline{\text{EOUT}}$	**Empty Output:** When active (Low), this output indicates that the 8X355 is empty, and further read operations cannot be accepted until at least one write operation is executed. $\overline{\text{EOUT}}$ is asserted in three situations: on power up, when $\overline{\text{FIN}}$ is inhibited, and when a read is performed with the stack pointer set at 0. In a cascaded configuration, $\overline{\text{EOUT}}$ is tied to $\overline{\text{EIN}}$ of the next lower LIFO chip of the stack.
4	$\overline{\text{EIN}}$	**Empty In:** In cascaded configurations, an active (Low) $\overline{\text{EIN}}$ signal indicates to a given LIFO chip that the next higher chip is empty. When $\overline{\text{EIN}}$ is inactive (High), the 8X355 will not respond to read operations; write operations are inhibited by $\overline{\text{FOUT}}$ (which is active), while selects and deselects are allowed. In non-cascaded configurations, $\overline{\text{EIN}}$ is usually forced low to enable reads and writes.
5	$\overline{\text{RESET}}$	**Reset:** When active (Low), this input returns the 8X355 to empty, deselected status. In cascaded configurations, RESETs are normally tied together.
6	$\overline{\text{FS}}$	**Fast Select:** When active (Low), this input temporarily allows read and write operations without performing a full select cycle.
7	$\overline{\text{ME}}$	**Master Enable:** When active (Low), this input indicates that the controller bank is active; otherwise, the 8X355 will not respond to any signal on the $\overline{\text{IV}}$ bus.
8	MCLK	**Master Clock:** A standard 8X355 clock input used for timing reference and system synchronization.
9	SC	**Select Command:** When active (High), this input indicates that data on the $\overline{\text{IV}}$ bus is to be interpreted as an address.
10	GND	**Ground**
11	WC	**Write Command:** When active (High), this signal indicates that data is being input from the $\overline{\text{IV}}$ bus.
12 to 19	$\overline{\text{IV0}}$ to $\overline{\text{IV7}}$	**Interface Vector Bus:** Active Low, bidirectional, 3-State signals which communicate I/O data and addresses.
20	V_{CC}	**Supply Voltage**

FUNCTIONAL OPERATION

The 8X355 performs two fundamental operations: write (push) and read (pop). These functions, along with the cascading capabilities and control logic of the device, are described below. Typical timing relationships for write, read, and select cycles are shown later.

Control Logic

Select Cycle — Before read or write operations can be executed, the 8X355 must be selected by either a standard select cycle or by Fast Select operation. With MCLK High, information entered on the $\overline{IV}$ bus is either an address or input data, depending on the states of SC and WC. An active (High) SC signal indicates that $\overline{IV}$ bus data is to be interpreted as an address; this address is then compared with that of the LIFO chip (130_8). If identical, the select latch becomes active (High) and the chip may be accessed for reads and writes. As shown in Table 1, the 8X355 remains in the selected state until another select cycle is initiated (MCLK = S – C = [High]), at which point the latch is updated.

Fast Select — A Fast Select ($\overline{FS}$) pin is provided which allows the 8X355 to be accessed whether or not a select cycle has been executed with the proper address. Since $\overline{FS}$ is ORed with the output of the select latch, this input has no effect on latch operation. Note that the 8X355 is selected only while $\overline{FS}$ is Low, and does not remain selected when $\overline{FS}$ goes High unless a proper select cycle had been performed during the time $\overline{FS}$ was Low. When coupled with external logic, this feature makes it possible to quickly push or pop data without using an extra cycle to select the stack peripheral.

Addressing — The 8X355 has a dedicated port address; however, the IV bus pins can be scrambled to obtain different addresses on a given bank. For example, a port designed with the address 130_8 (01011000_2) can be made to respond to the address 230_8 (10011000_2) simply by swapping the connections of the two most significant $\overline{IV}$ bus pins on a LIFO chip. Besides changing the enabling address, this pin swap has no effect on any other chip operation.

Reset — A reset pin is provided which empties and deselects the 8X355. When activated (Low), the $\overline{RESET}$ signal sets the stack pointer to 0, asserts $\overline{EOUT}$ (Low), negates $\overline{FOUT}$ (High), and negates select latch output.

Write/Read Operations

As shown in Table 1, the select latch, $\overline{IV}$ bus, and stack data are controlled by $\overline{ME}$, WC, SC, MCLK, $\overline{FS}$ and the current state of the select latch.

To perform a read, the 8X355 must be selected, and MCLK, SC, WC and $\overline{ME}$ must be low; in cascaded configurations, the 8X355 must also conform to status signal requirements (see Cascading Capabilities below). With these conditions satisfied, data is output on the $\overline{IV}$ bus and the stack pointer is decremented on the rising edge of MCLK.

A write is performed when MCLK and WC are High, $\overline{ME}$ and SC are Low, and the 8X355 is selected. When these conditions (and any status signal conditions) are satisfied, the stack pointer is incremented on the rising edge of the MCLK, after which data is loaded onto the top of the stack.

In single configurations, $\overline{FIN}$ and $\overline{EIN}$ are usually forced low (grounded) to enable reads and writes. In cascaded configurations, the $\overline{FIN}$ signal of the lowest LIFO chip and the $\overline{EIN}$ signal of the highest LIFO chip are similarly grounded.

Cascading Capabilities

To facilitate cascading, the 8X355 provides pins to detect and send signals indicating full or empty status. As described in PACKAGE AND PIN DESIGNATIONS, active $\overline{FOUT}$ and $\overline{EOUT}$ signals indicate, respectively, full and empty status. $\overline{FIN}$ is then defined by $\overline{FOUT}$ of the LIFO chip below, and $\overline{EIN}$ is defined as $\overline{EOUT}$ of the LIFO chip above. Thus data is stacked continuously from bottom to top unless provisions are made with external logic and Fast Select.

Empty status signals are updated on the rising edge of MCLK, while full status signals are updated on the falling edge. As described in Table 2, these signals control read and write operations of the 8X355 — note that LIFO chip #2 in the given configuration is used as the point of reference.

Table 1. General Operations

CONDITIONS						RESULTS		
$\overline{ME}$	SC	WC	MCLK	$\overline{FS}$	Select Latch	$\overline{IV}$ Bus	Data	Select Latch
H	X	X	X	X	X	Ignore	Keep	Keep
L	L	L	L	L	X^1	Output	Pop	Keep
L	L	L	L	H	L	Ignore	Keep	Keep
L	L	L	L	H	H^1	Output	Pop	Keep
L	L	L	H	X	X	Not Defined		
L	L	H	L	X	X	Ignore	Keep	Keep
L	L	H	H	L	X^2	Input	Push	Keep
L	L	H	H	H	L	Ignore	Keep	Keep
L	L	H	H	H	H^2	Input	Push	Keep
L	H	L	L	X	X	Ignore	Keep	Keep
L	H	L	H	X	X	Input (Address)	Keep	Update
X	H	H	X	X	X	Not Defined		

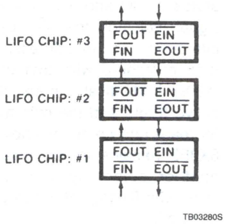
NOTES:
1. If not empty
2. If not full
X Don't Care

LIFO Stack Memory (32 × 8) 8X355

Table 2. Status Signal Operations in a Typical Cascaded Configuration

CONDITIONS				RESULTS				COMMENTS
$\overline{\text{EIN}}$	$\overline{\text{FIN}}$	$\overline{\text{EOUT}}$	$\overline{\text{FOUT}}$	$\overline{\text{EOUT}}$	$\overline{\text{FOUT}}$	Push	Pop	
X	X	L	L	Not Possible				Empty and full status
L	L	L	H	Update	H	Yes	No	Top of data stack at top of #1
L	L	H	L	H	Update	No	Yes	Top of data stack at top of #2
L	L	H	H	Update	Update	Yes	Yes	Top of data stack within #2
X	H	X	X	L	H	No	No	Top of data stack below #2
H	L	X	H	Not Defined				#2 not full and data in #3
H	L	H	L	H	L	No	No	Top of data stack above #2

NOTES:
Status signals and Push/Pop operations reference LIFO chip #2.
X = Don't Care



ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7.0	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_O	Off-state voltage	+5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to 150	°C

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-Pin plastic DIP	N8X355N
20-Pin ceramic DIP	N8X355F

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%/0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	TEST CONDITIONS[1]	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				0.8	V
V_{OH}	High-level output voltage	$\overline{IV}$ pins: V_{CC} = Min/I_{OH} = −3.2mA	2.4			V
V_{OL}	Low-level output voltage	$\overline{IV}$ pins: V_{CC} = Min/I_{OL} = 16mA $\overline{EOUT}$ & $\overline{FOUT}$: V_{CC} = Min/I_{OL} = 8mA			0.55 0.55	V V
V_{IC}	Input clamp voltage	V_{CC} = Min/I_{IN} = −10mA			−1	V
I_{IH}	High-level input current	All inputs except $\overline{IV}$ pins: V_{CC} = Max/V_{IN} = 4.5V			100	μA
I_{IL}	Low-level input current	All inputs except $\overline{IV}$ pins: V_{CC} = Max/V_{IN} = 0.5V			−100	μA
I_{OS}	Short circuit output current[2]	$\overline{IV}$ pins: V_{CC} = Max	−30		−140	mA
I_{CEX}	High state leakage current	$\overline{EOUT}$ & $\overline{FOUT}$ = High; V_{CC} = Min/V_O = 5.5V			100	μA
I_{OZH}	High-Z state output current (high level)	$\overline{IV}$ pins: V_{CC} = Max/V_O = 2.7V			100	μA
I_{OZL}	High-Z state output current (low level)	$\overline{IV}$ pins: V_{CC} = Max/V_O = 0.5V			−100	μA
I_{CC}	Supply current	All inputs except $\overline{IV}$ pins: V_{CC} = Max/V_{IN} = 0V			200	mA

NOTES:

1. All voltages measured with respect to ground terminal.
2. At any one time, no more than one output should be connected to ground.

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V \pm 5\%/0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS[1, 2, 3]	LIMITS		UNITS
		From	To		Min	Max	
Pulse Widths							
t_{WMC}	MCLK pulse width	↑ MCLK	↓ MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	35		ns
t_{WRS}	Reset pulse width[4]	↓ $\overline{RESET}$	↑ $\overline{RESET}$	$\overline{FS}$ = High; $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	60		ns
Setup Times							
t_{SIVMC}	$\overline{IV}$ setup time	$\overline{IV}$	↓ MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	40		ns
t_{SWCMC}	Write command setup time	↑ WC	↓ MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; SC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	60		ns
t_{SSCMC}	Select command setup time	↑ SC	↓ MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; WC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	55		ns
t_{SMEMC}	Master enable setup time	↓ $\overline{ME}$	↓ MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{EIN}$ & $\overline{FIN}$ = Low	55		ns
Hold Times							
t_{HIVMC}	$\overline{IV}$ hold time	↓ MCLK	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	5		
t_{HMEMC}	Master enable hold time	↓ MCLK	↑ $\overline{ME}$	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{EIN}$ & $\overline{FIN}$ = Low	5		ns
t_{HSCMC}	Select command hold-time	↓ MCLK	↓ SC	$\overline{FS}$ & $\overline{RESET}$ = High; WC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	0		
t_{HWCMC}	Write command hold time	↓ MCLK	↓ WC	$\overline{FS}$ & $\overline{RESET}$ = High; SC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	2		ns

AC ELECTRICAL CHARACTERISTICS (Continued)

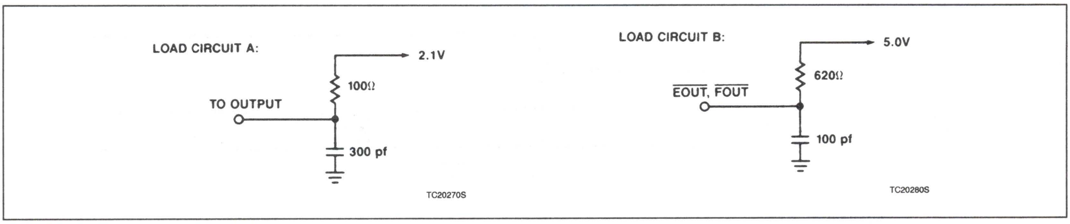
SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS[1, 2, 3]	LIMITS		UNITS
		From	To		Min	Max	
Enable/Disable Times							
t_{EIVWC}	$\overline{IV}$ output enable time	↓ WC	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; SC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low		30	ns
t_{EIVSC}	$\overline{IV}$ output enable time	↓ SC	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low		30	ns
t_{EIVME}	$\overline{IV}$ output enable time	↓ $\overline{ME}$	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, SC, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low		30	ns
t_{EIVFS}	$\overline{IV}$ output enable time	↓ $\overline{FS}$	$\overline{IV}$	$\overline{RESET}$ = High, WC, SC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low		35	ns
t_{DIVWC}	$\overline{IV}$ output disable time	↑ WC	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; SC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low[5]		25	ns
t_{DIVSC}	$\overline{IV}$ output disable time	↑ SC	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low[5]		25	ns
t_{DIVME}	$\overline{IV}$ output disable time	↑ $\overline{ME}$	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, SC, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low[5]		25	ns
t_{DIVFS}	$\overline{IV}$ output disable time	↑ $\overline{FS}$	$\overline{IV}$	$\overline{RESET}$ = High; WC, SC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low[5]		30	ns
Propagation Delays							
t_{PEOMC}	Empty output propagation delay; both High-to-Low and Low-to-High transitions	↑ MCLK	$\overline{EOUT}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, SC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low		40	ns
t_{tPFOMC}	Full output propagation delay; both High-to-Low and Low-to-High transitions	↓ MCLK	$\overline{FOUT}$	$\overline{FS}$ & $\overline{RESET}$ = High; SC $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low		45	ns
Stack Timing							
t_{NOPOP}	Maximum spurious read (or pop) time[6]	↓ SC or ↓ WC	↑ $\overline{ME}$ or ↑ $\overline{FS}$	MCLK = Low		15	ns
t_{POP}	Minimum read (or pop) time[7]	Setup read condition (see truth table)	Stack popped	SC, WC & MCLK = Low; or $\overline{ME}$ or $\overline{FS}$ = Low	40		ns

NOTES:

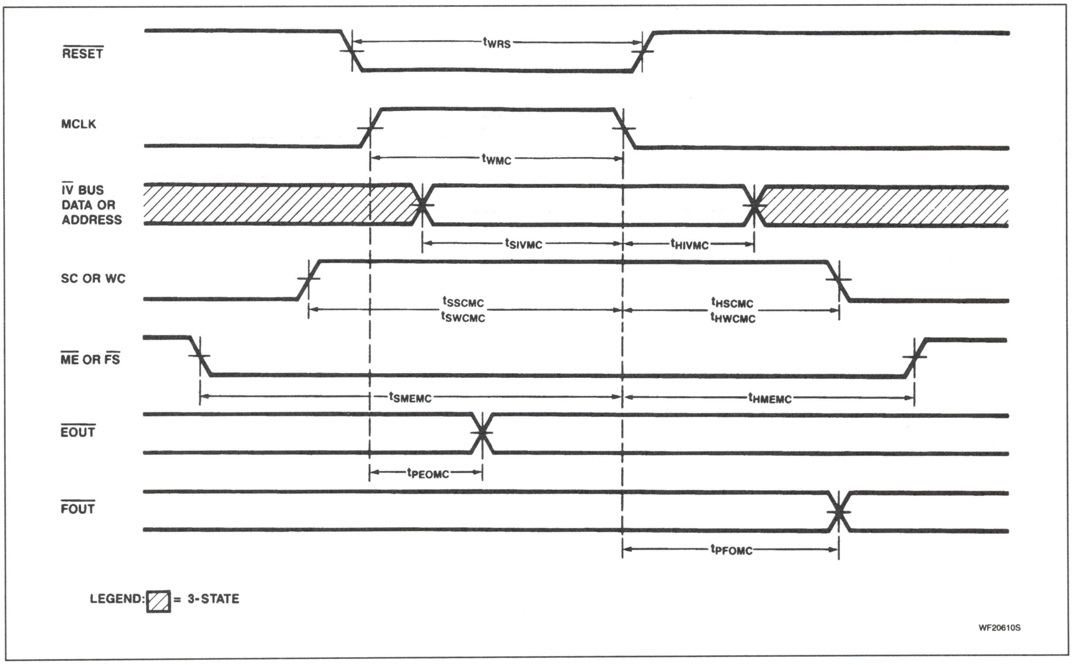
1. Loading — see TEST LOADING circuits.
2. All inputs are driven between 0V and 3V volts; input timing parameters are measured at 1.5V.
3. Except for output disable times, all output timing parameters are measured at 1.5V.
4. Resetting the part initializes $\overline{EOUT}$ and $\overline{FOUT}$, respectively, to low and high states and 3-States the $\overline{IV}$ bus; all of these conditions occur within the minimum $\overline{RESET}$ pulse width.
5. These parameters are measured with a capacitive loading of 50pF and represent the output driver turn-off time. Disable times measured into a capacitive loading of 300pF do not exceed 40ns except for $\overline{FS}$ which is 45ns.
6. This parameter represents the maximum time a Read (Pop) condition can be logically asserted, that is, as a decoding transient, without popping the stack.
7. This parameter represents the minimum time a Read (Pop) condition must be logically asserted to pop the stack.

LIFO Stack Memory (32 × 8) 8X355

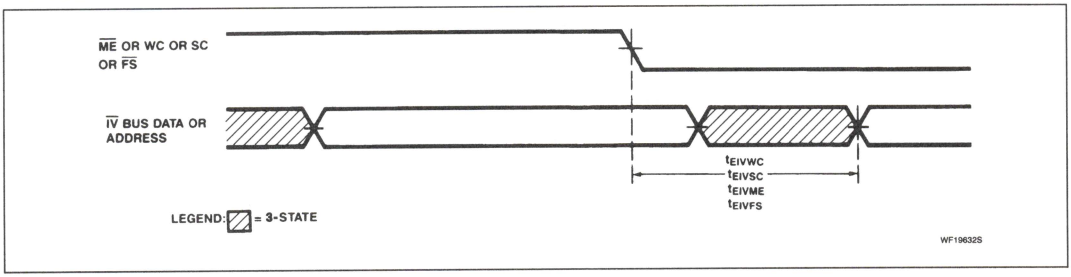
TEST LOADING CIRCUITS


LOAD CIRCUIT A:
2.1V
100Ω
TO OUTPUT
300 pf
TC20270S
LOAD CIRCUIT B:
5.0V
620Ω
EOUT, FOUT
100 pf
TC20280S

OVERALL TIMING


RESET
tWRS
MCLK
tWMC
IV BUS DATA OR ADDRESS
tSIVMC
tHIVMC
SC OR WC
tSSCMC
tSWCMC
tHSCMC
tHWCMC
ME OR FS
tSMEMC
tHMEMC
EOUT
tPEOMC
FOUT
tPFOMC
LEGEND: ▨ = 3-STATE
WF20610S

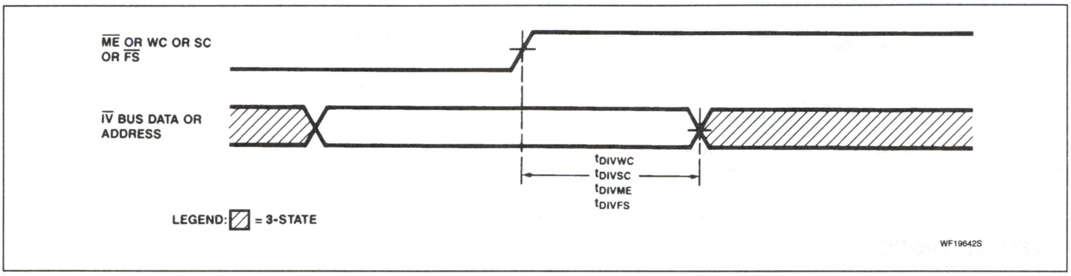
ENABLE TIMING


ME OR WC OR SC OR FS
IV BUS DATA OR ADDRESS
tEIVWC
tEIVSC
tEIVME
tEIVFS
LEGEND: ▨ = 3-STATE
WF19632S

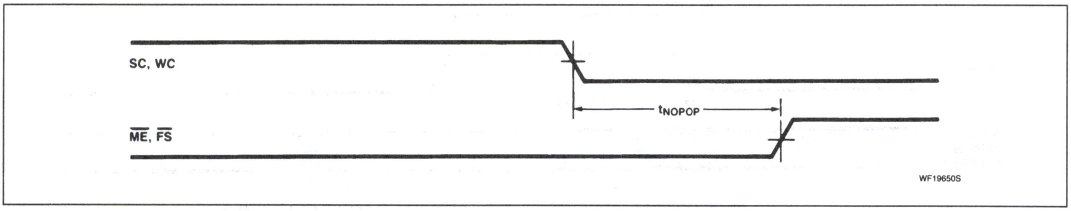
LIFO Stack Memory (32 × 8)

8X355

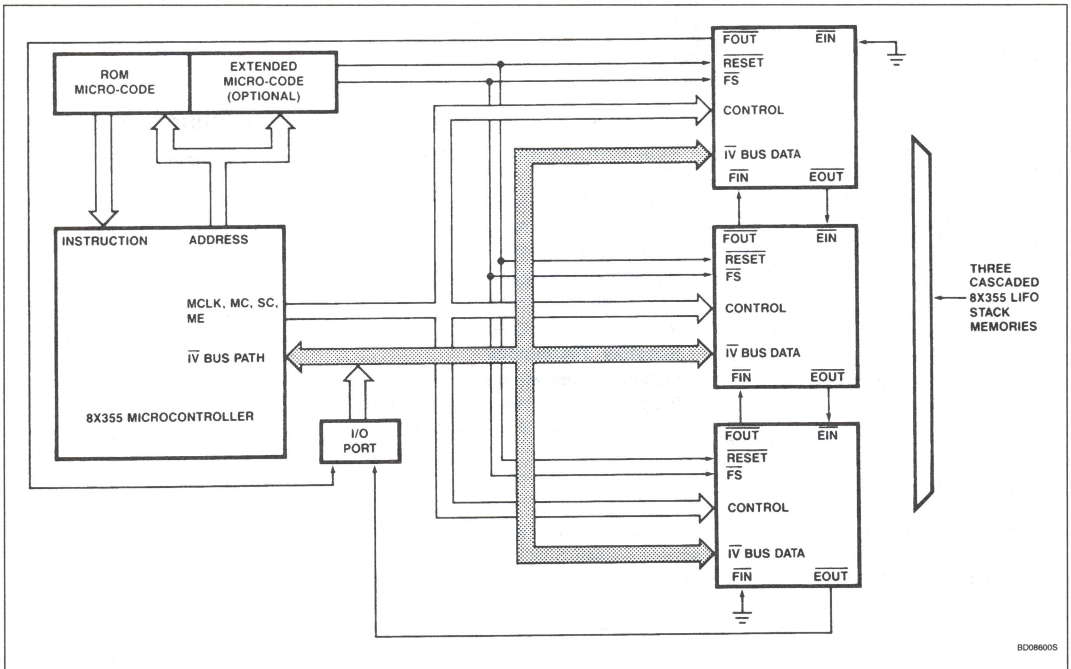
DISABLE TIMING


ME OR WC OR SC OR FS
IV BUS DATA OR ADDRESS
tDIVWC
tDIVSC
tDIVME
tDIVFS
LEGEND: ▨ = 3-STATE
WF19642S

STACK TIMING DIAGRAM


SC, WC
ME, FS
tNOPOP
WF19650S

IMPLEMENTATION OF CASCADED 8X355's


ROM MICRO-CODE
EXTENDED MICRO-CODE (OPTIONAL)
INSTRUCTION
ADDRESS
MCLK, MC, SC, ME
$\overline{IV}$ BUS PATH
8X355 MICROCONTROLLER
I/O PORT
$\overline{FOUT}$
$\overline{EIN}$
$\overline{RESET}$
$\overline{FS}$
CONTROL
$\overline{IV}$ BUS DATA
$\overline{FIN}$
$\overline{EOUT}$
THREE CASCADED 8X355 LIFO STACK MEMORIES
BD08600S

NOTE:
In this configuration, the extended microcode is normally latched on the falling edge of MCLK to ensure validity of the microcode throughout the write cycle.

Signetics

8X360
Memory Address Director (MAD)

Product Specification

Microprocessor Products

DESCRIPTION

The 8X360 Memory Address Director (MAD) (Figure 1), is a high-speed, performance oriented peripheral. The 8X360 reduces bookkeeping and software overhead in memory intensive systems. In a system context, the 8X360 interfaces directly with the 8X305 Microcontroller. The MAD can address up to 64K of memory and provides an efficient, cost-effective solution to applications requiring large memories and high-speed data transfers.

The 8X360 must be initialized before the registers/counters can begin data transfer. The eleven registers/counters are initialized with information such as starting/ending address, byte count, address increment/decrement, etc. The registers/counters are addressable through a select instruction from the 8X305 (conventional) or by extended microcode.

FUNCTIONAL OPERATION

The 8X360 has 11 registers/counters of which two, the Control and Status Register and Frequency Divider, determine the mode of operation.

The 8X360 wakes up with three registers/counters: Frequency Divide; Control and Status Register; and Word Counter in a defined state. All others wake up in an undefined state (see Table 3). No address is selected when the 8X360 wakes up.

Registers

The address registers appear as two 8-bit addressable registers to the 8X305 and are initialized with a starting address. The word count register (two 8-bit addressable registers) holds a 16-bit value for either the final address or the word/byte count, depending on the mode of operation. Loading the counters from these registers is non-destructive.

FEATURES

- **Automatic Memory Address Management**
- **16-Bit Address Capability**
- **11 Addressable Registers for Maximum Flexibility**
- **Directly Compatible with the 8X305 Microcontroller**
- **Bipolar ISL and Low-Power Schottky Technology**
- **Multiple Clock Modes with Programmable Frequency Divider Terminal Count End-of-Block Indicator**
- **40-Pin Dual In-Line Package**

Counters

The 16-bit address and word counters appear as two 8-bit addressable registers. The address counter counts up or down from the current address by 1 or 2, with its output available on 16 3-State output pins which are enabled through the TSCL pin. The word counter counts down from an initialized number of bytes or words, or stores the final address.

I/O Register

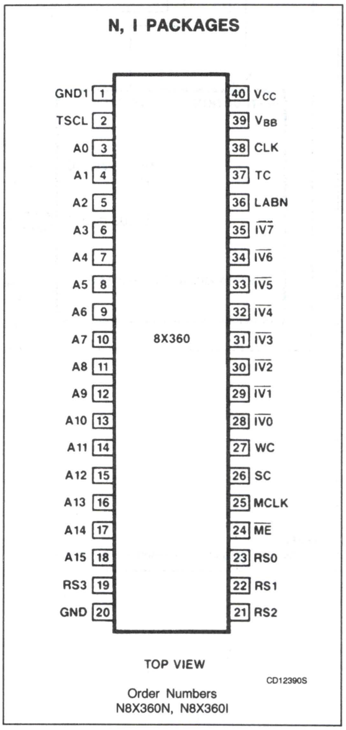
The I/O register stores a special 8-bit address from 0-255. In the I/O clocking mode, an address on the $\overline{IV}$ bus and the stored address are compared after the rising edge of MCLK. A match produces an internal clock which clocks the 8X360 counters.

STATUS REGISTERS

Command Status

The CSR is an 8-bit addressable register with its most significant bit (bit 0) not used. The other seven bits control the overall operations, such as: the address counter step length and direction, the mode of operation, the counter load and enable functions, the loop abort and terminal count actions (see Table 1 for a summary of this Register.)

PIN CONFIGURATION


N, I PACKAGES
GND1 1
TSCL 2
A0 3
A1 4
A2 5
A3 6
A4 7
A5 8
A6 9
A7 10
A8 11
A9 12
A10 13
A11 14
A12 15
A13 16
A14 17
A15 18
RS3 19
GND 20
40 V_{CC}
39 V_{BB}
38 CLK
37 TC
36 LABN
35 $\overline{IV7}$
34 $\overline{IV6}$
33 $\overline{IV5}$
32 $\overline{IV4}$
31 $\overline{IV3}$
30 $\overline{IV2}$
29 $\overline{IV1}$
28 $\overline{IV0}$
27 WC
26 SC
25 MCLK
24 $\overline{ME}$
23 RS0
22 RS1
21 RS2
8X360
TOP VIEW
CD12390S

Order Numbers
N8X360N, N8X360I

Memory Address Director (MAD) 8X360

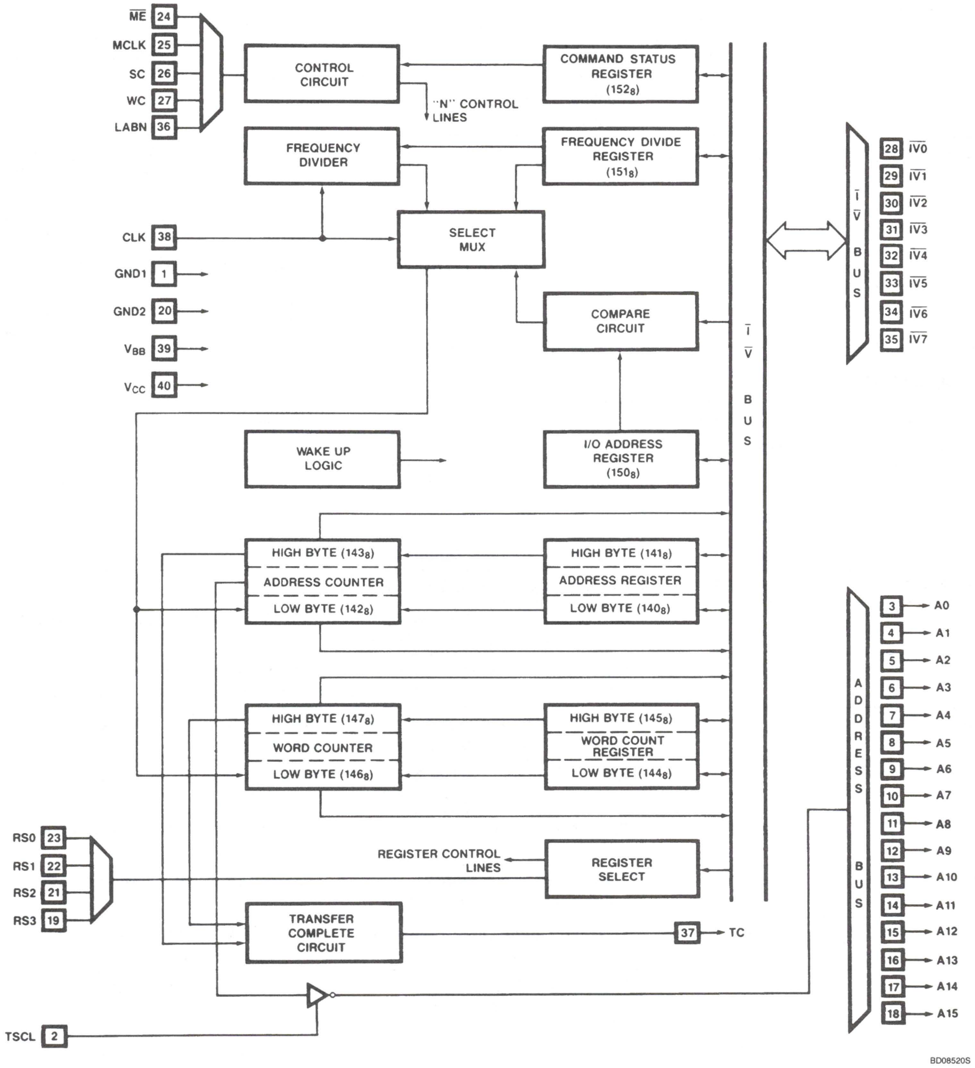

ME 24
MCLK 25
SC 26
WC 27
LABN 36
CONTROL CIRCUIT
COMMAND STATUS REGISTER (152₈)
"N" CONTROL LINES
FREQUENCY DIVIDER
FREQUENCY DIVIDE REGISTER (151₈)
CLK 38
SELECT MUX
GND1 1
GND2 20
VBB 39
VCC 40
COMPARE CIRCUIT
WAKE UP LOGIC
I/O ADDRESS REGISTER (150₈)
IV BUS
28 IV0
29 IV1
30 IV2
31 IV3
32 IV4
33 IV5
34 IV6
35 IV7
HIGH BYTE (143₈)
ADDRESS COUNTER
LOW BYTE (142₈)
HIGH BYTE (141₈)
ADDRESS REGISTER
LOW BYTE (140₈)
HIGH BYTE (147₈)
WORD COUNTER
LOW BYTE (146₈)
HIGH BYTE (145₈)
WORD COUNT REGISTER
LOW BYTE (144₈)
RS0 23
RS1 22
RS2 21
RS3 19
REGISTER CONTROL LINES
REGISTER SELECT
TRANSFER COMPLETE CIRCUIT
37 TC
TSCL 2
ADDRESS BUS
3 A0
4 A1
5 A2
6 A3
7 A4
8 A5
9 A6
10 A7
11 A8
12 A9
13 A10
14 A11
15 A12
16 A13
17 A14
18 A15
BD08520S

Figure 1. Block Diagram

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
40-Pin plastic DIP	N8X360N
40-Pin ceramic DIP	N8X360I

Table 1. Command Status Register

	BIT 0	BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7
Function	Not used	Address counter step length	Address counter counting direction	Mode of operation	Load counters	Counter enable bit	Loop abort bit	Terminal count bit
Programming Level 1	Not used	Count by 2	Count down	Address compare	Load the counters	Enable the counters	Loop has been aborted, counting inhibited	Data transfer is complete, counting is inhibited
0	Not used	Count by 1	Count up	Word count	No action	Counters are disabled	Loop is not aborted	Data transfer is not complete

PIN DESCRIPTION

PIN NO	IDENTIFIER	FUNCTION
1, 20	GND1, GND2	Ground
2	TSCL	**A**ddress **3**-State **C**ontrol **P**in **TSCL** — **Action** 1 — Enable A0 – A15 0 — Disable (3-State)
3 – 18	A0 – A15	**16 3-State Address Output Pins:** A0 is LSB.
19, 21 – 23	RS3 – RS0	**Register Select Input Pins:** RS0 is LSB.
24	$\overline{ME}$	**Master Enable:** Active low input to enable the $\overline{IV}$ bus for data input, data output, or $\overline{IV}$ Address Selection/Deselection.
25	MCLK	**Master Clock:** Active high input (from the Microcontroller) used to strobe data into the registers of the 8X360 from the $\overline{IV}$ bus and synchronize $\overline{IV}$ address selection.
26	SC	**Select Command:** Active high input (from Microcontroller) to enable $\overline{IV}$ bus address input from the $\overline{IV}$ bus for register/counter selection.
27	WC	**Write Command:** Active high input (from Microcontroller) to enable the writing of data into the registers from $\overline{IV}$ bus.
28 – 35	$\overline{IV0}$ – $\overline{IV7}$	**Eight 3-State Bidirectional $\overline{IV}$ Bus Pins:** Data to and from 8X305, active low: $\overline{IV7}$ is LSB.
36	LABN	**Loop Abort Input Signal:** The rising edge of this signal inhibits counting and sets the loop abort bit in the CSR.
37	TC	**Terminal Count Output:** Active when the contents of the address counter compare with the word counter or when the word counter reaches zero.
38	CLK	**External Input Clock Pin:** An externally-generated, positive-going signal to clock the address and word counters.
39	V_{BB}	Supply
40	V_{CC}	Supply

Bits 1 and 2 (Address Counter Step Length and Direction) — Bit 1 selects a counter step length of 1 or 2 when programmed to a logic 0 or 1, respectively. Bit 2 determines the address counter direction, a logic 0 for up or a logic 1 for down.

Bit 3 (Mode of Operation) — There are two modes of operation defined in Bit 3, the address compare or the word count. A logic 1 selects the address compare mode and a logic 0 selects the word count mode. The compare mode stores a final address in the word counter. A match between the address counter and the word counter inhibits the address counter from counting further. The word count mode initializes the word counter with a specific number of bytes to decrement from. Counting stops when the word counter reaches zero.

Bits 4 and 5 (Load and Enable) — Bit 4 at a logic 1 loads the contents of the address and word registers to their respective counters. (External clock should be in low-state when loading.) Bit 4 at a logic 0 has no action. Bit 4 is reset automatically when loading is completed. Bit 5 enables the counters to counting when programmed to a logic 1 and Bits 6 and 7 are at a logic 0. (See Bits 6 & 7 in the following paragraphs.) A logic 0 in Bit 5 disables the counters.

Bit 6 (Loop Abort Bit) — Bit 6 goes high when the LABN pin goes high, inhibiting counting. Counting resumes when the counters are enabled and LABN has been cleared (Bit 6 is low).

Bit 7 (Terminal Count Bit) — Bit 7 is a read-only bit, reflecting the status of the TC pin. The terminal count bit goes high at the end of a data transfer operation, thus inhibiting further counting. Bit 7 is cleared when the word counter is loaded with a non-zero value, not equal to the contents of the address counter in the address compare mode.

Frequency Divide Register

The Frequency Divide Register is an 8-bit addressable register which selects the clocking source and programs the frequency divider. The most significant two bits, bits 0 and 1, select the clocking source; external clock, external clock with frequency divide, or special I/O mode. The least significant six bits contain the frequency divisor, a number from 2 to 64 which is entered as the divisor (N) minus one (N – 1), and is loaded into the Frequency Divider.

The divisor controls the number of clock pulses which occur before the counters change state one time. Loading to the Frequency Divider is non-destructive. (See Table 2 for a summary of this register.)

Table 2. Frequency Divide Register

FREQUENCY DIVIDE REGISTER BITS			FUNCTION
Bit 0	Bit 1	Bits 2 – 7	
0	0	Not used	External Clock
0	1	Frequency Divider	External Clock with Frequency Divider
1	1	Not used	Special I/O Address Circuitry
1	0	Not used	Not used

Table 3. Read/Write Summary With Register Wake-Up States

8X360 REGISTER NAME	READ	WRITE	WAKE-UP STATE (OCTAL)
Address Register Low = 140_8 High = 141_8	X	X	Not defined
Address Counter Low = 142_8 High = 143_8	X	No	Not defined
Word Count Register Low = 144_8 High = 145_8	X	X	Not defined
Word Counter Low = 146_8 High = 147_8	X	No	000
Input/Output Special Register	X	X	Not defined
Frequency Divide Register	X	X	Bit 0 = 0 Bit 1 = 0
Command and Status Register	X	X	001

NOTES:
X = Operation can be done.
No = Operation cannot be done.

Transfer Complete Circuit

The transfer complete circuit (TC) signals the end of data block transfer or end of count.

Wake Up Circuit

The wake up circuit wakes up the Word Counter, the Frequency Divide Register, bits 0 and 1, and the CSR in a zero state, with the exception of bit 7 of the CSR, which wakes up in a 1 state. All other registers/counters wake up in an undefined state. (See Table 3 for Register Wake Up States.)

Special I/O Compare Circuit

The compare circuit is used when the Special I/O clocking mode is selected by the two most significant bits of the Frequency Divide Register. This circuit performs the comparison between the address on the $\overline{\text{IV}}$ bus and the address stored in the Special I/O register. The comparison is done after the rising edge of MCLK. A match generates an internal clock pulse, causing the counters to change state one time.

Clocking Modes

The 8X360 has three clocking modes available to the user. Each clocking mode facilitates a different need. The Special I/O mode can toggle the counters when a preassigned address appears on the $\overline{\text{IV}}$ bus. The external clock is used whenever the Special I/O mode is not used. The external clock can be used in conjunction with a programmable Frequency Divide Register. The frequency divide mode regulates the number of clock pulses necessary to clock the counters once. (See Figure 2.)

Special I/O Mode — The Special I/O register is initialized with an address. When the address appears on the $\overline{\text{IV}}$ bus, it is compared with the preassigned address; a match produces a clock pulse, no match leaves the 8X360 unchanged. (See Flowchart 1.)

External Clock — The External Clock can be connected to MCLK (synchronous) or to any other user source (synchronous or asynchronous). The asynchronous clock should be a positive going clock and should rest in the low state. The clock pulse will change the state of the address and word counters one time. (See Flowchart 1.)

Frequency Divide — The Frequency Divider is used with the external clock, and can be initialized with a number from 2 to 64, from the Frequency Divide Register. The actual number loaded is the divisor minus one, which determines the clock pulses necessary

Memory Address Director (MAD) 8X360

for the Address and Word Counters to toggle one time (See Flowchart 1).

$\overline{IV}$ Bus

The input/output of the $\overline{IV}$ bus is controlled by SC, WC, MCLK, and $\overline{ME}$. SC is activated by the 8X305 Microcontroller and enables the 8X360 for addressing of the internal registers from the $\overline{IV}$ bus. A different $\overline{IV}$ address deselects the previously selected register. WC is a write strobe asserted by the microcontroller which enables writing into the 8X360. MCLK is an active-high input from the microcontroller which strobes data into the 8X360 and provides $\overline{IV}$ address selection. $\overline{ME}$ is a Master Enable input which is usually connected to either $\overline{LB}$ or $\overline{RB}$ outputs of the 8X305 (Table 4).

LABN

The **L**oop **A**bort pi**n** (LABN) is rising edge sensitive and asynchronous. LABN disables the counters when high. Counting resumes from last count (unless reinitialized), when LABN is low and the counters are enabled by the CSR.

Addressing Techniques

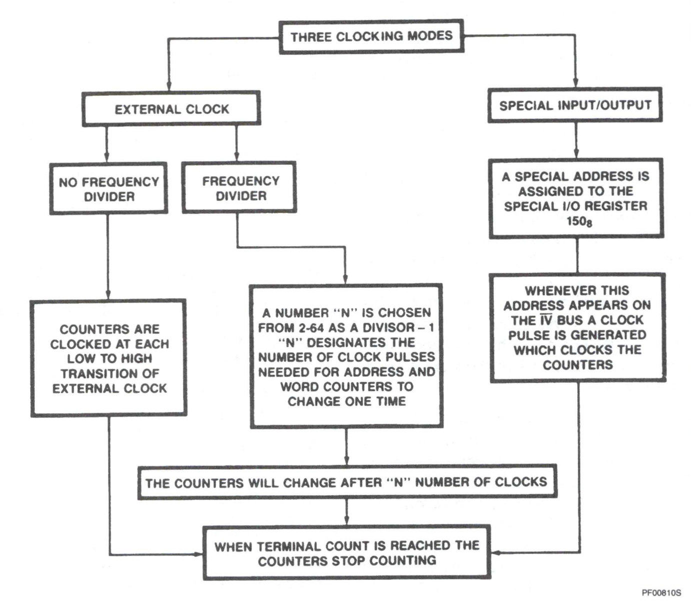
The addressable registers in the 8X360 can be selected through either conventional $\overline{IV}$ select or extended microcode.

$\overline{IV}$ Bus Select (Conventional Select)

When a register is initially selected for a read or write, two 8X305 instructions are needed. The first instruction selects a register and the second instruction performs the actual read or write to the 8X360. Subsequent read or write operations to the same register can then be performed without additional select instructions (Note: Pins RS0 – RS3 = 1).

Extended Microcode

The extended microcode method can select and read/write to a register in one 8X305 instruction cycle. The 8X360 registers are selected by the ROM's extended microcode bits through pins RS0 – RS3 (See Table 5 for Register Addresses).


THREE CLOCKING MODES
EXTERNAL CLOCK
SPECIAL INPUT/OUTPUT
NO FREQUENCY DIVIDER
FREQUENCY DIVIDER
A SPECIAL ADDRESS IS ASSIGNED TO THE SPECIAL I/O REGISTER 150₈
WHENEVER THIS ADDRESS APPEARS ON THE IV BUS A CLOCK PULSE IS GENERATED WHICH CLOCKS THE COUNTERS
COUNTERS ARE CLOCKED AT EACH LOW TO HIGH TRANSITION OF EXTERNAL CLOCK
A NUMBER "N" IS CHOSEN FROM 2-64 AS A DIVISOR – 1 "N" DESIGNATES THE NUMBER OF CLOCK PULSES NEEDED FOR ADDRESS AND WORD COUNTERS TO CHANGE ONE TIME
THE COUNTERS WILL CHANGE AFTER "N" NUMBER OF CLOCKS
WHEN TERMINAL COUNT IS REACHED THE COUNTERS STOP COUNTING
PF00810S

Flowchart 1

Table 4. IV Bus Control States

SC	WC	MCLK	$\overline{ME}$	ACTION
H	L	H	L	8X360 is addressed
L	H	H	L	Data is written to the 8X360
L	L	X	L	8X360 registers are read

Table 5. Register Addresses

REGISTER	ADDRESS	ADDRESSES FOR THE RS PINS			
		RS3	RS2	RS1	RS0
Address Register LSB	140_8	0	0	0	0
Address Register MSB	141_8	0	0	0	1
Address Counter LSB	142_8	0	0	1	0
Address Counter MSB	143_8	0	0	1	1
Word Count Register LSB	144_8	0	1	0	0
Word Count Register MSB	145_8	0	1	0	1
Word Counter LSB	146_8	0	1	1	0
Word Counter MSB	147_8	0	1	1	1
Special I/O Address Register	150_8	1	0	0	0
Frequency Divider Register	151_8	1	0	0	1
Command and Status Register	152_8	1	0	1	0

Memory Address Director (MAD) 8X360

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $0°C \leqslant T_A \leqslant 70°C$, $4.75V \leqslant V_{CC} \leqslant 5.25V$, $1.7V \leqslant V_{BB} \leqslant 2.1V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{CC}	Supply		4.75	5	5.25	V
V_{BB}	Supply		1.7	1.9	2.1	V
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				0.8	V
V_{CL}	Input clamp diode voltage	V_{CC} = Min, I_1 = −10mA			−1.5	V
I_{IH}	High-level input current	V_{CC} = Max, I_{IH} = 2.7V			100	µA
I_{IL}	Low-level input current	V_{CC} = Max, V_{IL} = 0.5V			−500	µA
V_{OL}	Low-level output voltage	V_{CC} = Min				
	$\overline{IV}$	I_{OL} = 16mA			0.55	V
	TC	I_{OL} = 8mA			0.55	V
	A0 – A15	I_{OL} = 8mA			0.55	V
V_{OH}	High-level output voltage	V_{CC} = Min				
	$\overline{IV}$	I_{OH} = −3.2mA	2.4			V
	TC	I_{OH} = −3.2mA	2.4			V
	A0 – A15	I_{OH} = −3.2mA	2.4			V
I_{OS}	Short circuit output current	V_{CC} = Max				
	$\overline{IV}$	Only 1 output at a time	−20			mA
	TC		−20			mA
	A0 – A15		−20			mA
I_{CC}		V_{CC} = Max		70	100	mA
I_{BB}		V_{BB} = 1.9V	180	265	350	mA

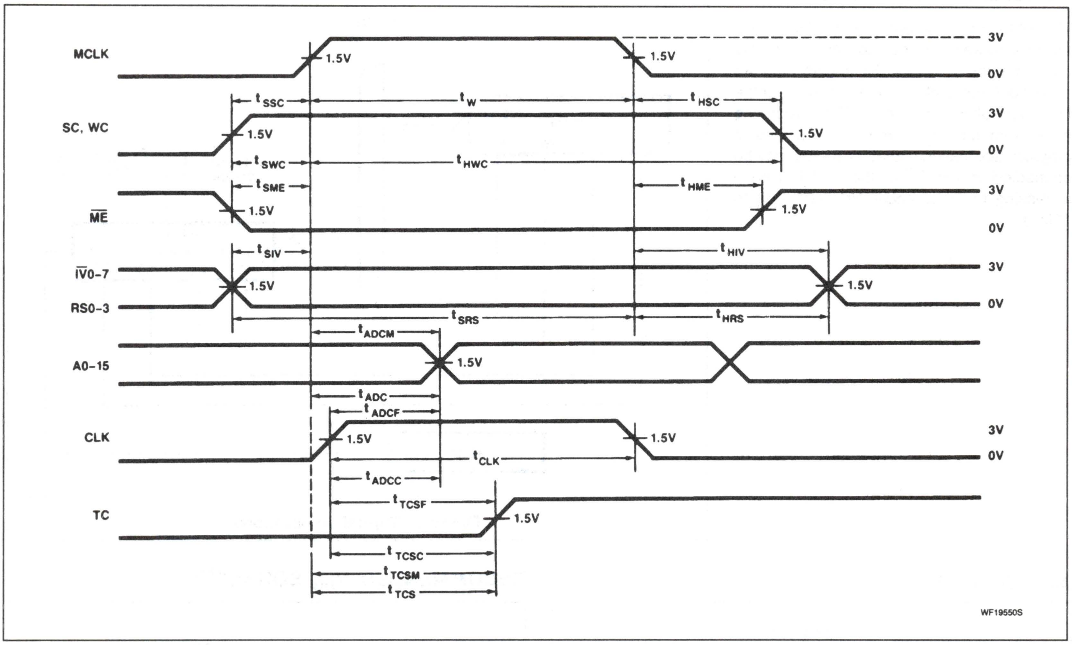
AC ELECTRICAL CHARACTERISTICS $4.75V \leqslant V_{CC} \leqslant 5.25V$, $1.7V \leqslant V_{BB} \leqslant 2.1V$, $0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS		UNIT
		From	To		Min	Max	
Pulse Widths							
t_{CLK}	CLOCK pulse width	CLK1 ↑	CLK ↓		25		ns
t_W	MCLK pulse width	MCLK ↑	MCLK ↓		35		ns
t_{LABN}	LABN pulse width	LABN ↑	LABN ↓		8		ns
Propagation Delays **External Clock Mode**							
t_{ADCC}	Address bus delay	CLK ↑	A0 – A15 ↑ ↓		60		ns
t_{TCSC}	TC set delay	CLK ↑	TC ↑		70		ns
Frequency Divide Mode							
t_{ADCF}	Address bus delay	CLK ↑	A0 – A15 ↑ ↓		90		ns
t_{TCSF}	TC set delay	CLK ↑	TC ↑		100		ns
Special I/O Mode							
t_{ADCM}	Address bus delay	MCLK ↑	A0 – A15 ↑ ↓		95		ns
t_{TCSM}	TC set delay	MCLK ↑	TC ↑		115		ns
Delays For READ/WRITE Operations							
t_{ADC}	Address bus delay	MCLK ↑	A0 – A15 ↑ ↓		85		ns
t_{TCS}	TC set delay	MCLK ↑	TC ↑		105		ns
Setup Times							
t_{SLA}	LABN setup time	LABN ↑	CLK ↑		20		ns
t_{SRS}	RS setup time	RS ↑ ↓	MCLK ↓		15		ns
t_{SIV}	$\overline{\text{IV}}$ setup time	$\overline{\text{IV}}$ ↑ ↓	MCLK ↑		0		ns
t_{SWC}	WC setup time	WC ↑	MCLK ↑		20		ns
t_{SSC}	SC setup time	SC ↑	MCLK ↑		32		ns
t_{SME}	$\overline{\text{ME}}$ setup time	$\overline{\text{ME}}$ ↓	MCLK ↑		0		ns
Hold Times							
t_{WRS}	RS hold time	MCLK ↓	RS ↑		15		ns
t_{HIV}	$\overline{\text{IV}}$ hold time	MCLK ↓	$\overline{\text{IV}}$ ↓		0		ns
t_{HWC}	WC hold time	MCLK ↑	WC ↓		37		ns
t_{HSC}	SC hold time	MCLK ↓	SC ↓		0		ns
t_{HME}	ME hold time	MCLK ↓	$\overline{\text{ME}}$ ↑		0		ns
Enable Times							
t_{ESCL}	Address bus enable time	TSCL ↑	A0 – A15 ↑ ↓			25	ns
t_{OEME}	$\overline{\text{IV}}$ bus enable time	$\overline{\text{ME}}$ ↓	$\overline{\text{IV}}$ Bus ↑ ↓			30	ns
t_{OESC}	$\overline{\text{IV}}$ bus enable time	SC ↓	$\overline{\text{IV}}$ Bus ↑ ↓			30	ns
t_{OEWC}	$\overline{\text{IV}}$ bus enable time	WC ↓	$\overline{\text{IV}}$ Bus ↑ ↓			30	ns
Disable Times (see note 1)							
t_{OSCL}	Address bus disable time	TSCL ↓	A0 – A15 ↑ ↓			35	ns
t_{ODME}	$\overline{\text{IV}}$ bus disable time	$\overline{\text{ME}}$ ↑	$\overline{\text{IV}}$ Bus ↑ ↓	(1)		25	ns
t_{ODSC}	$\overline{\text{IV}}$ bus disable time	SC ↑	$\overline{\text{IV}}$ Bus ↑ ↓	(1)		25	ns
t_{ODWC}	$\overline{\text{IV}}$ bus disable time	WC ↑	$\overline{\text{IV}}$ Bus ↑ ↓	(1)		25	ns

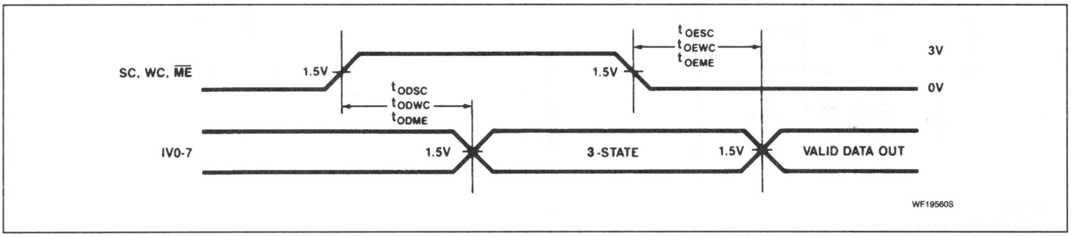
NOTE:

1. These parameters are measured with a capacitive loading of 50pF and represent the output driver turn-off time. Disable times measured into a capacitive loading of 300pF will be a maximum of 35ns.

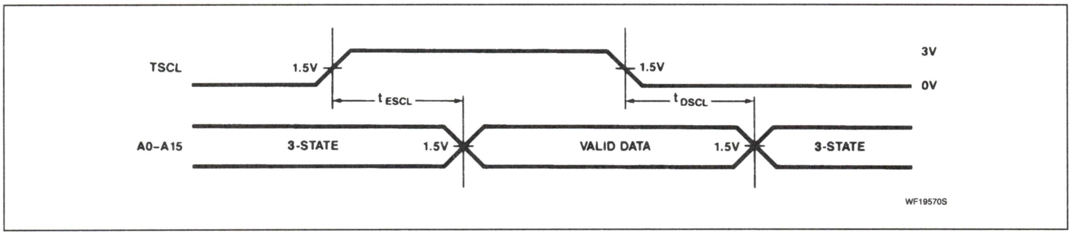
SETUP, HOLD, AND PROPAGATION DELAY


MCLK
SC, WC
ME
IV0–7
RS0–3
A0–15
CLK
TC
1.5V
3V
0V
tSSC
tW
tHSC
tSWC
tHWC
tSME
tHME
tSIV
tHIV
tSRS
tHRS
tADCM
tADC
tADCF
tCLK
tADCC
tTCSF
tTCSC
tTCSM
tTCS
WF19550S

IV ENABLE AND DISABLE


SC, WC, ME
IV0-7
1.5V
3V
0V
tOESC
tOEWC
tOEME
tODSC
tODWC
tODME
3-STATE
VALID DATA OUT
WF19560S

ADDRESS ENABLE/DISABLE


TSCL
A0–A15
1.5V
3V
0V
tESCL
tDSCL
3-STATE
VALID DATA
3-STATE
WF19570S

Memory Address Director (MAD) 8X360

TYPICAL APPLICATIONS

The 8X360 has 16 address output lines which enables addressing of up to 64K of memory on each bank of the 8X360. (Figure 2 shows one 8X360 connected to the Left Bank of the Microcontroller. For applications requiring more than 64K of addressing, such as in graphics applications, another 8X360 may be connected to the $\overline{RB}$ of the 8X305. Hence increasing the total addressing range up to 128K.)

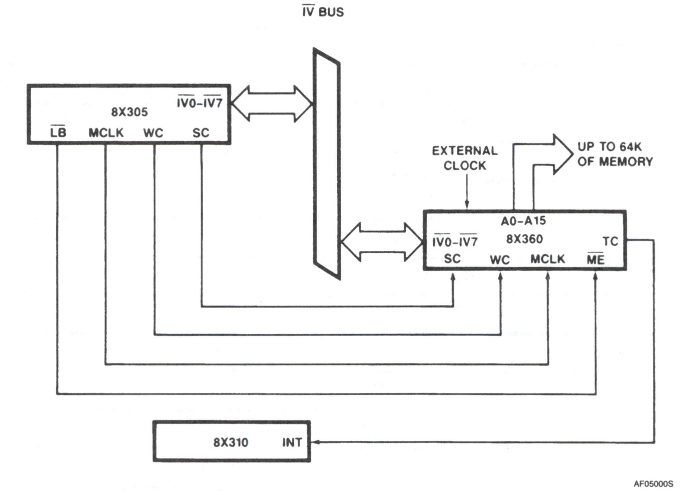
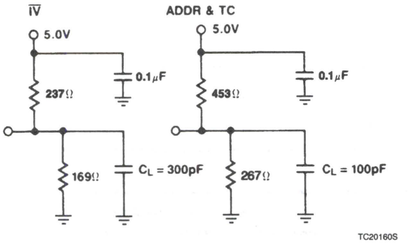

$\overline{IV}$ BUS
8X305
$\overline{IV0}$–$\overline{IV7}$
$\overline{LB}$
MCLK
WC
SC
EXTERNAL CLOCK
UP TO 64K OF MEMORY
A0–A15
$\overline{IV0}$–$\overline{IV7}$
8X360
TC
SC
WC
MCLK
$\overline{ME}$
8X310
INT
AF05000S

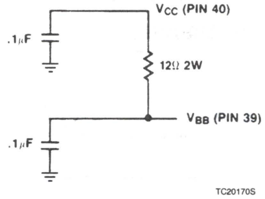
Figure 2. Typical Applications

AC LOADS(1)


$\overline{IV}$
5.0V
0.1μF
237Ω
169Ω
C_L = 300pF
ADDR & TC
5.0V
0.1μF
453Ω
267Ω
C_L = 100pF
TC20160S

NOTES:
1. 1μF capacitors are for decoupling only.
2. 12Ω resistor value derived using typical I_{BB} value of 265mA at 25°C and 5V V_{CC}.

RECOMMENDED V_{BB} SOURCE(1, 2)


V_{CC} (PIN 40)
.1μF
12Ω 2W
V_{BB} (PIN 39)
.1μF
TC20170S

Signetics

8X371

8-Bit Latched Bidirectional I/O Port

Product Specification

Microprocessor Products

DESCRIPTION

The 8X371 I/O Port is a bidirectional device designed for use as an interface element in systems that use TTL-compatible busses. Typically, the 8X371 is used with the 8X305 Microcontroller and its associated Interface Vector ($\overline{\text{IV}}$) bus; however, it can also be used with the 8X300 Microcontroller or an equivalent microprocessor. The 8X371 is functionally the same and pin-for-pin compatible with the older 8T31/8X31 but features improved performance and increased drive current. As shown in the logic diagram of Figure 1, the 8X371 consists of eight identical data latches — bits 0 through 7. The latches are accessed from either of two 8-bit busses — the Microcontroller ($\overline{\text{IV}}$ bus) and the user data (UD bus). Separate controls are provided for each bus and both busses operate independently, except when both attempt to input data at the same time; in such situations, the user bus always has priority. A Master Enable ($\overline{\text{ME}}$) input is available for additional control over the $\overline{\text{IV}}$ bus. The data latches are transparent, in that, while either bus is enabled for input, all input-data transitions are propagated to the other bus, if enabled for output.

ORDERING INFORMATION

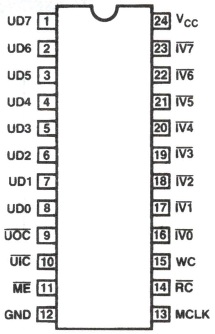
DESCRIPTION	ORDER CODE
24-Pin plastic DIP	N8X371N
24-Pin ceramic DIP	N8X371I

FEATURES

- **Two bidirectional 8-bit busses**
- **Independent bus operation (user-bus priority for data entry)**
- **User data input synchronous with respect to MCLK**
- **3-State TTL outputs with high-drive capabilities**
- **Power-up to predetermined logic state**
- **Directly compatible with 8X305 (or 8X300) Microcontrollers**
- **Single +5V supply**
- **0.4″ 24-pin DIP**

PIN CONFIGURATION

N, I, PACKAGES


UD7 1
UD6 2
UD5 3
UD4 4
UD3 5
UD2 6
UD1 7
UD0 8
$\overline{\text{UOC}}$ 9
$\overline{\text{UIC}}$ 10
$\overline{\text{ME}}$ 11
GND 12
24 V_{CC}
23 $\overline{\text{IV7}}$
22 $\overline{\text{IV6}}$
21 $\overline{\text{IV5}}$
20 $\overline{\text{IV4}}$
19 $\overline{\text{IV3}}$
18 $\overline{\text{IV2}}$
17 $\overline{\text{IV1}}$
16 $\overline{\text{IV0}}$
15 WC
14 $\overline{\text{RC}}$
13 MCLK

TOP VIEW

CD12370S

PIN. NO.	IDENTIFIER	FUNCTION
1 – 8	UD7 – UD0	3-State, bidirectional User Data (UD) bus; UD0 and corresponds to $\overline{\text{IV0}}$.
9	$\overline{\text{UOC}}$	User Output Control — active low input to enable data output to UD0 – UD7.
10	$\overline{\text{UIC}}$	User Input Control — active low input to enable data input from UD0 – UD7.
11	$\overline{\text{ME}}$	Master Enable — active low input to enable the $\overline{\text{IV}}$ bus for data input, or data output; UD-bus operations are unaffected.
12	GND	Ground.
13	MCLK	Master Clock — active high input (from Microcontroller used to strobe data into data latches from the $\overline{\text{IV}}$ and UD buses.
14	$\overline{\text{RC}}$	Read Control — active low input to enable data output to $\overline{\text{IV0}}$ – $\overline{\text{IV7}}$.
15	WC	Write Command — active high input (from Microcontroller) to enable the writing of data into the data latches from the $\overline{\text{IV}}$ bus (provided $\overline{\text{UIC}}$ is not low).
16 – 23	$\overline{\text{IV0}}$ – $\overline{\text{IV7}}$	Interface Vector (Input/Output Bus) — 3-State, bidirectional, Microcontroller data bus; $\overline{\text{IV0}}$ corresponds to UD0.
24	V_{CC}	+5V power supply.

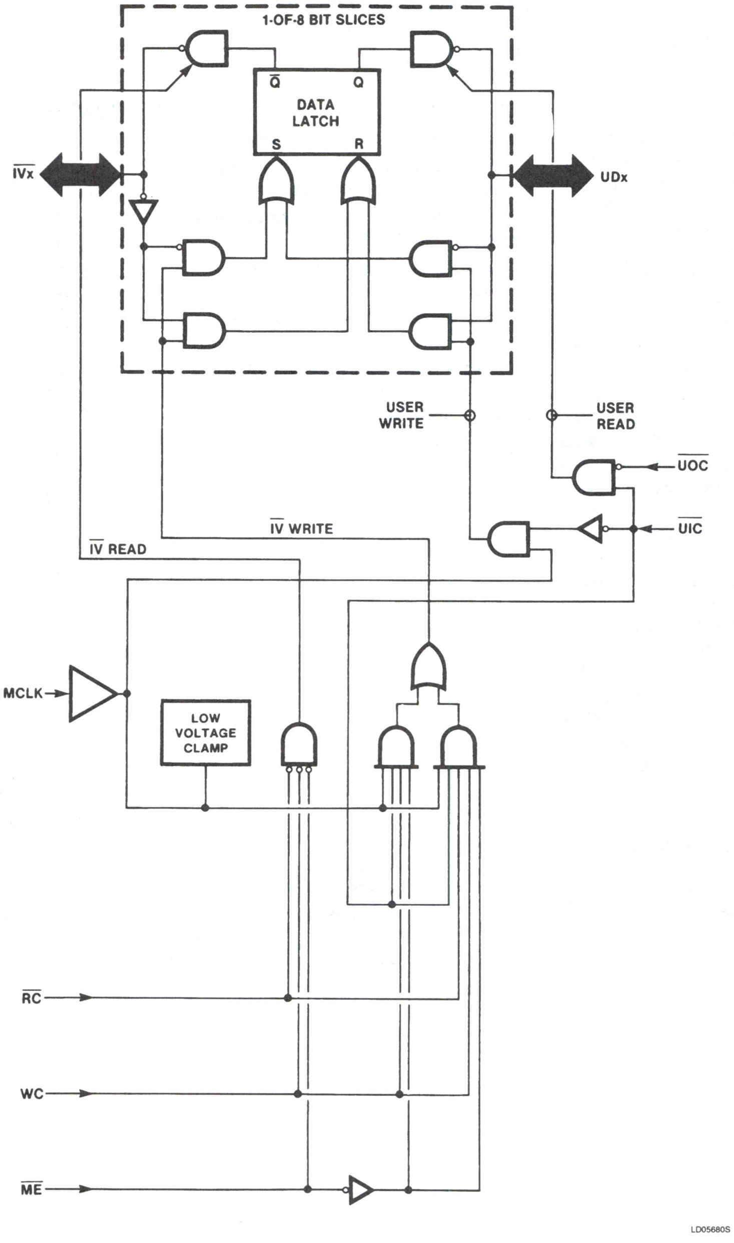

1-OF-8 BIT SLICES
$\overline{Q}$
Q
DATA LATCH
S
R
$\overline{IVx}$
UDx
USER WRITE
USER READ
$\overline{UOC}$
$\overline{UIC}$
$\overline{IV}$ WRITE
$\overline{IV}$ READ
MCLK
LOW VOLTAGE CLAMP
$\overline{RC}$
WC
$\overline{ME}$
LD05680S

Figure 1. Logic Diagram for 8X371 I/O Port

8-Bit Latched Bidirectional I/O Port 8X371

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data (UD) bus interface is controlled by the $\overline{\text{UIC}}$ and $\overline{\text{UOC}}$ inputs. Data input to the UD bus is synchronous with MCLK, that is, with $\overline{\text{UIC}}$ low, information is written into the data latches only when MCLK is high. Output drivers on the UD bus are enabled when $\overline{\text{UOC}}$ is low and $\overline{\text{UIC}}$ is high.

Table 1. Input/Output Control of UD Bus

$\overline{\text{UIC}}$	$\overline{\text{UOC}}$	MCLK	FUNCTION OF UD BUS
H	L	X	Output data
L	X	H	Input data
L	X	L	Inactive
H	H	X	Inactive

NOTE:
X = don't care

$\overline{\text{IV}}$ Bus Control

Input/output control of the $\overline{\text{IV}}$ bus is shown in Table 2; this bus is controlled by $\overline{\text{RC}}$, WC, $\overline{\text{ME}}$, and MCLK. The $\overline{\text{IV}}$ bus is enabled for output (Microcontroller read operation) when $\overline{\text{ME}}$, $\overline{\text{RC}}$, and WC are all low. Data is written into the data latches from the $\overline{\text{IV}}$ bus when $\overline{\text{ME}}$ is low and both WC and MCLK are high. To avoid data-input conflicts, inputs from the $\overline{\text{IV}}$ bus are inhibited when $\overline{\text{UIC}}$ is low; under all other conditions, the $\overline{\text{IV}}$ and UD busses operate independently. The Microcontroller Left Bank $(\overline{\text{LB}})$ and Right Bank $(\overline{\text{RB}})$ outputs can control the $\overline{\text{ME}}$ inputs for two banks of I/O devices, thus acting as a ninth address bit. If more than one I/O Port (including the addressable parts — 8X372, 8X376, 8X382, etc.) are to be connected to the same bank ($\overline{\text{LB}}$ or $\overline{\text{RB}}$) of the Microcontroller, selection of each 8X371 must be accomplished with external logic to avoid bus conflicts.

Table 2. Input/Output Control of $\overline{\text{IV}}$ Bus

$\overline{\text{ME}}$	$\overline{\text{RC}}$	WC	MCLK	$\overline{\text{UIC}}$	FUNCTION OF $\overline{\text{IV}}$ BUS
L	L	L	X	X	Output Data
L	X	H	H	H	Input Data
L	H	L	X	X	Inactive
L	X	H	X	L	Inactive
L	X	H	L	H	Inactive
H	X	X	X	X	Inactive

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note: *A logic "1" in Microcontroller software corresponds to a high-level on the UD bus even though the $\overline{\text{IV}}$ bus is inverted.*) The 8X371 wakes up in the unselected state with all data bits latched at the "logic 1" level (UD) bus outputs high if enabled).

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7	V_{DC}
V_I	Input voltage	+5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $4.75 \leqslant V_{CC} \leqslant 5.25V$, $0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS Min	Typ	Max	UNIT
V_{CC}	Supply voltage		4.75	5	5.25	V
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				0.8	V
V_{IK}	Input clamp voltage	V_{CC} = Min; $I_I = -10mA$			−1.5	V
I_{IH}	High-level input current[1]	V_{CC} = Max; $V_{IH} = 2.7V$		5	100	μA
I_{IL}	Low-level input current[1]	V_{CC} = Max; $V_{IL} = 0.5V$		−350	−550	μA
V_{OL}	Low-level output voltage $\overline{\text{IV}}$ bus ($\overline{\text{IV0}}$ – $\overline{\text{IV7}}$)	V_{CC} = Min; $I_{OL} = 16mA$			0.55	V
	User bus (UD4 – UD7)	V_{CC} = Min; $I_{OL} = 24mA$			0.55	V
V_{OH}	High-level output voltage	V_{CC} = Min; $I_{OH} = -3.2mA$	2.4			V
I_{OS}	Short circuit output current[2] $\overline{\text{IV}}$ bus ($\overline{\text{IV0}}$ – $\overline{\text{IV7}}$)	V_{CC} = Max	−20			mA
	UD bus (UD4 – UD7)	V_{CC} = Max	−10			mA
I_{CC}	Supply current	V_{CC} = Max; $\overline{\text{ME}} = \overline{\text{UOC}} = V_{CC}$		90	150	mA

NOTES:
1. The input current includes the 3-State leakage current of the output driver on the data lines.
2. Only one output may be shorted at a time.

8-Bit Latched Bidirectional I/O Port

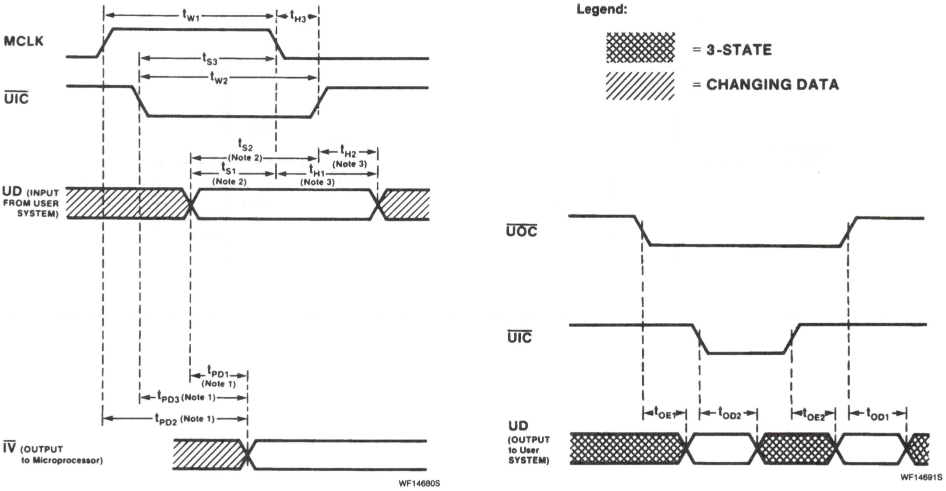
8X371

AC ELECTRICAL CHARACTERISTICS $4.75 \leqslant V_{CC} \leqslant 5.25V$, $0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS			UNIT
		From	To		Min	Typ	Max	
Pulse Widths								
t_{W1}	Clock high	↑ MCLK	↓ MCLK		35			ns
t_{W2}	User input control	↓ $\overline{UIC}$	↑ $\overline{UIC}$	MCLK = High	35			ns
Propagation Delays								
t_{PD1}	UD propagation delay	UD	$\overline{IV}$	MCLK = High $\overline{RC}$ = WC = $\overline{ME}$ = $\overline{UCI}$ = Low			30	ns
t_{PD2}	UD clock delay	↑ MCLK	$\overline{IV}$	UD = Stable; $\overline{RC}$ = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			50	ns
t_{PD3}	UD input delay	↓ $\overline{UIC}$	$\overline{IV}$	UD = Stable; MCLK = High $\overline{RC}$ = WC = $\overline{ME}$ = Low			50	ns
t_{PD4}	$\overline{IV}$ data propagation delay	$\overline{IV}$	UD	MCLK = WC = $\overline{UIC}$ = High; $\overline{ME}$ = $\overline{UOC}$ = $\overline{RC}$ = Low			45	ns
t_{PD5}	$\overline{IV}$ data clock delay	↑ MCLK	UD	WC = $\overline{UIC}$ = High; $\overline{IV}$ = Stable $\overline{ME}$ = $\overline{UOC}$ = $\overline{RC}$ = Low			55	ns
Output Enable Timing								
t_{OE1}	UD output enable	↓ UOC	UD	$\overline{UIC}$ = High			30	ns
t_{OE2}	UD input recovery	↑ $\overline{UIC}$	UD	$\overline{UOC}$ = Low			30	ns
t_{OE3}	$\overline{IV}$ data master enable	↓ $\overline{ME}$	$\overline{IV}$	WC = $\overline{RC}$ = Low			22	ns
t_{OE4}	$\overline{IV}$ data read enable	↓ $\overline{RC}$	$\overline{IV}$	WC = $\overline{ME}$ = Low			25	ns
t_{OE5}	$\overline{IV}$ data write recovery	↓ WC	$\overline{IV}$	$\overline{RC}$ = $\overline{ME}$ = Low			25	ns
Output Disable Timing								
t_{OD1}	UD output disable	↑ $\overline{UOC}$	UD	$\overline{UIC}$ = High			25	ns
t_{OD2}	UD input override	↓ $\overline{UIC}$	UD	$\overline{UOC}$ = Low			30	ns
t_{OD3}[1]	$\overline{IV}$ data master disable	↑ $\overline{ME}$	$\overline{IV}$	WC = $\overline{RC}$ = Low			20	ns
t_{OD4}[1]	$\overline{IV}$ data read disable	↑ $\overline{RC}$	$\overline{IV}$	WC = $\overline{ME}$ = Low			20	ns
t_{OD5}[1]	$\overline{IV}$ data write override	↑ WC	$\overline{IV}$	$\overline{RC}$ = $\overline{ME}$ = Low			20	ns
Setup Time								
t_{S1}	UD clock setup time	UD	↓ MCLK	$\overline{UIC}$ = Low	15			ns
t_{S2}	UD setup time	UD	↑ UIC	MCLK = High	15			ns
t_{S3}	User input control setup time	↓ $\overline{UIC}$	↓ MCLK		25			ns
t_{S4}	$\overline{IV}$ data setup time	$\overline{IV}$	↓ MCLK	WC = $\overline{UIC}$ = High; $\overline{ME}$ = Low	35			ns
t_{S5}[2]	$\overline{IV}$ master enable setup time	↓ $\overline{ME}$	↓ MCLK	WC = $\overline{UIC}$ = High	30			ns
t_{S6}	$\overline{IV}$ write control setup time	↑ WC	↓ MCLK	$\overline{ME}$ = Low; $\overline{UIC}$ = High	30			ns
Hold Times								
t_{H1}	UD clock hold time	↓ MCLK	UD	$\overline{UIC}$ = Low	15			ns
t_{H2}	UD control hold time	↑ $\overline{UIC}$	UD	MCLK = High	15			ns
t_{H3}	User input control hold time	↓ MCLK	↑ UIC		0			ns
t_{H4}	$\overline{IV}$ data hold time	↓ MCLK	$\overline{IV}$	WC = $\overline{UIC}$ = High; $\overline{ME}$ = Low	5			ns
t_{H5}[2]	$\overline{IV}$ master enable hold time	↓ MCLK	↑ $\overline{ME}$	WE = $\overline{UIC}$ = High	0			ns
t_{H6}	$\overline{IV}$ write control hold time	↓ MCLK	↓ WC	$\overline{ME}$ = Low; = $\overline{UIC}$ = High	0			ns

NOTES:

1. These parameters are measured with a capacitive loading of 50pf and represent the output driver turn-off time.
2. If $\overline{ME}$ is to be high (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing into or selection of the I/O port.

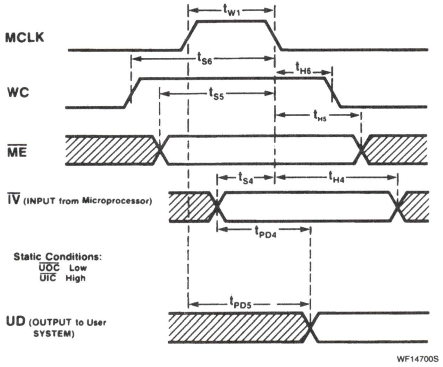

MCLK
UIC
UD (INPUT FROM USER SYSTEM)
IV (OUTPUT to Microprocessor)
tW1
tH3
tS3
tW2
tS2 (Note 2)
tH2 (Note 3)
tS1 (Note 2)
tH1 (Note 3)
tPD1 (Note 1)
tPD3 (Note 1)
tPD2 (Note 1)
WF14680S
Legend:
▩ = 3-STATE
▨ = CHANGING DATA
UOC
UIC
UD (OUTPUT to User SYSTEM)
tOE1
tOD2
tOE2
tOD1
WF14691S

STATIC CONDITIONS:
SC = WC = $\overline{ME}$ = Low; $\overline{UOC}$ = High

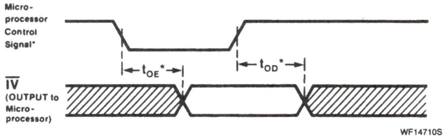
NOTES:
1. The actual time for stable data on the $\overline{IV}$ bus is the latest propagation from t_{PD1}, t_{PD2}, and t_{PD3}.
2. The UD input must satisfy the setup-time requirements for both t_{S1} and t_{S2}.
3. Minimum hold-time required for the UD input is the earlier of the times specified by t_{H1} and t_{H2}.

a. User Data Input Timing

b. User Data Output Timing


MCLK
WC
ME
IV (INPUT from Microprocessor)
UD (OUTPUT to User SYSTEM)
tW1
tS6
tH6
tS5
tH5
tS4
tH4
tPD4
tPD5
Static Conditions:
UOC Low
UIC High
WF14700S

c. Microcontroller Write Cycle Timing


Micro-processor Control Signal*
IV (OUTPUT to Micro-processor)
tOE*
tOD*
WF14710S

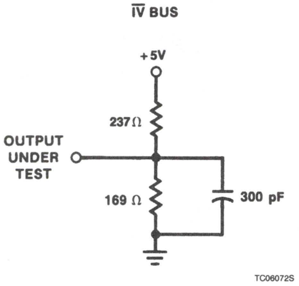
***PARAMETER KEY**

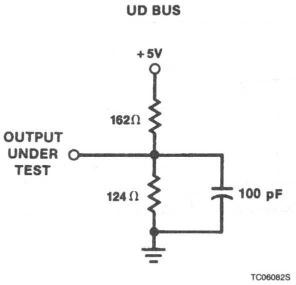
MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS	STATIC CONDITIONS
$\overline{ME}$	t_{OE3} t_{OD3}	SC = WC = Low
$\overline{RC}$	t_{OE4} t_{OD4}	SC = $\overline{ME}$ = Low
SC	t_{OE5} t_{OD5}	$\overline{RC}$ = $\overline{ME}$ = Low

d. Microcontroller Select Cycle Timing

Figure 2. Timing Diagram

AC TEST LOADING CIRCUITS


$\overline{IV}$ BUS
+5V
237 Ω
OUTPUT UNDER TEST
169 Ω
300 pF
TC06072S


UD BUS
+5V
162 Ω
OUTPUT UNDER TEST
124 Ω
100 pF
TC06082S

8-Bit Latched Bidirectional I/O Port 8X371

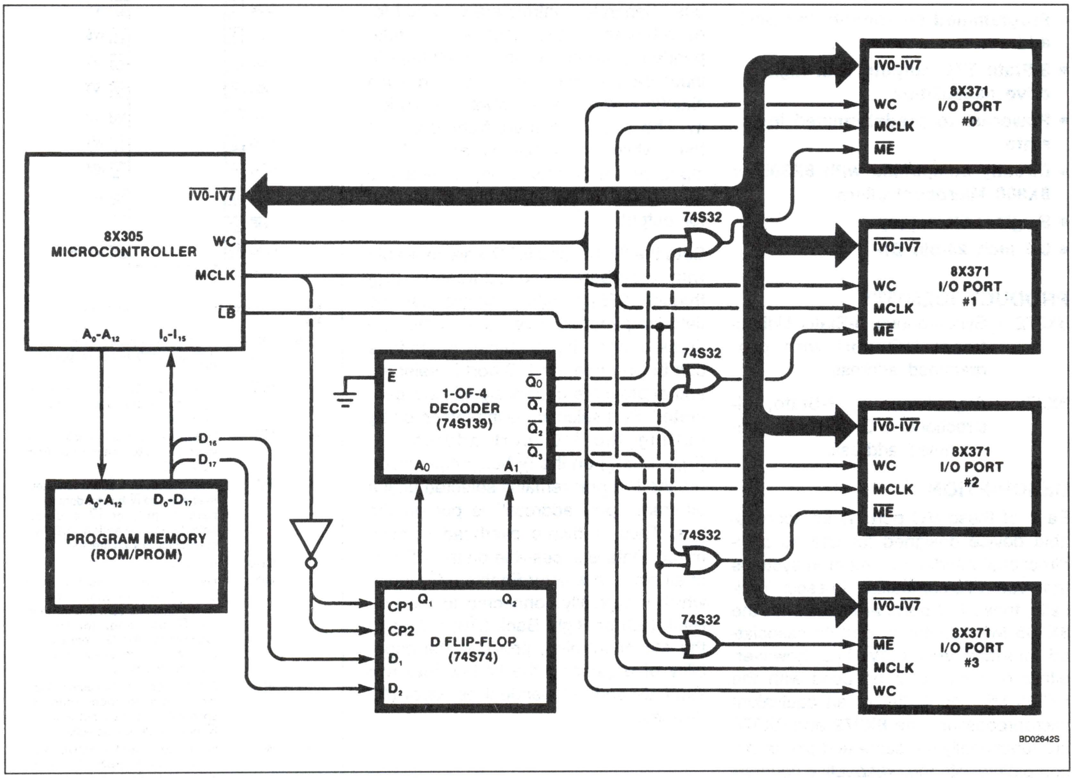
APPLICATIONS

In some applications, performance of a Microcontroller system can be enhanced by using the 8X371 I/O Port instead of an addressable 8X372 port. Using a technique referred to as Extended Microcode or Fast $\overline{\text{IV}}$ Select, the address select cycles which normally precede a read or write operation when using an 8X372 can be eliminated by use of the 8X371.

This technique is often used in bit slice microprocessor designs and involves widening the program memory beyond the normal 16-bit requirement of the Microcontroller. The extra bits are used as enable signals for the 8X371 ports. Thus, the 8X371 is enabled during the instruction cycle in which it is required for input/output operations. Since the software overhead of separate address select cycles is eliminated, the overall system performance is improved.

As shown in the accompanying diagram, the program memory is extended by two bit positions (D_{16} and D_{17}), permitting any one of four 8X371 ports to be enabled during those instructions that perform input/output operations. Because of timing considerations, latches must be used to hold the Extended Microcode through the end of the instruction cycle. A decoder is used to obtain four enable signals from the two extra bits. The decoder outputs are ORed with the $\overline{\text{LB}}$ output of the 8X305; thus, all four I/O ports are placed on the Left Bank of the $\overline{\text{IV}}$ bus.

I/O PORT SELECTION USING EXTENDED MICROCODE


IV0-IV7
8X371 I/O PORT #0
WC
MCLK
ME
IV0-IV7
8X305 MICROCONTROLLER
WC
MCLK
LB
A0-A12
I0-I15
74S32
IV0-IV7
8X371 I/O PORT #1
WC
MCLK
ME
74S32
E
1-OF-4 DECODER (74S139)
Q0
Q1
Q2
Q3
A0
A1
IV0-IV7
8X371 I/O PORT #2
WC
MCLK
ME
D16
D17
A0-A12
D0-D17
PROGRAM MEMORY (ROM/PROM)
74S32
CP1
Q1
Q2
CP2
D FLIP-FLOP (74S74)
D1
D2
74S32
IV0-IV7
8X371 I/O PORT #3
ME
MCLK
WC
BD02642S

Signetics

8X372/8X376
Addressable/Bidirectional I/O Ports

Product Specification

Microprocessor Products

FEATURES

- **Two bidirectional 8-bit busses**
- **Independent bus operation (user-bus priority for data entry)**
- **User data input synchronous (8X372) or asynchronous (8X376) with respect to MCLK**
- **Programmed Microcontroller port address**
- **3-State TTL outputs with high-drive capabilities**
- **Power-up to predetermined logic state**
- **Directly compatible with 8X305 or 8X300 Microcontrollers**
- **Single +5V supply**
- **0.4 inch 24-pin DIP**

PRODUCT IDENTITY

8X372 — Synchronous, 3-State, bidirectional I/O port with programmed address.

8X376 — Asynchronous, 3-State, bidirectional I/O port with programmed address.

DESCRIPTION

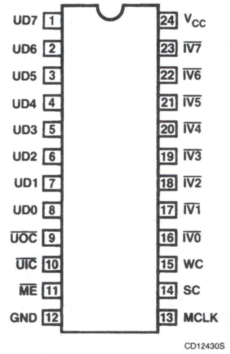
Each of these I/O ports is an addressable device designed for use as a bidirectional interface element in systems that use TTL-compatible busses. Typically, these I/O ports are used with the 8X305 Microcontroller and its associated Interface Vector ($\overline{\text{IV}}$) bus; however, either port can also be used with the 8X300 Microcontroller or an equivalent microprocessor. The 8X372 and 8X376 are functionally the same and pin-for-pin compatible with their respective counterparts, the 8T32/8X32 and 8T36/8X36; however, the new parts feature better performance, increased drive current, and improved programming procedures.

As shown in the logic diagram of Figure 1, each I/O port consists of eight identical data latches — bits 0 through 7. These latches are accessed through either of two 8-bit busses — one connecting to the Microcontroller ($\overline{\text{IV}}$ bus) and the other to the user system (UD bus). Separate controls are provided for each bus and both busses operate independently, except when both attempt to input data at the same time. In such situations, the user bus always has priority. The data latches are transparent, in that, while either bus is enabled for input, all transitions in input data are propagated to the other bus, if enabled for output.

Both the 8X372 and 8X376 are available with preprogrammed addresses (0_{10} through 255_{10}); either device can be field-programmed over the same address range. Input/output operations can begin once the I/O port is selected and appropriate control signals are generated. Port selection is implemented by putting the I/O port address (0_{10} – 255_{10}) on the $\overline{\text{IV}}$ bus; once selected, the I/O port remains selected until a different "port address" is put on the bus. Thus, software overhead is minimized. Data is accessible on the UD bus at all times. A Master Enable ($\overline{\text{ME}}$) input, which is typically connected to the Left Bank ($\overline{\text{LB}}$) or Right Bank ($\overline{\text{RB}}$) output of the Microcontroller, provides the capability of organizing the $\overline{\text{IV}}$ bus into two separate and independent banks of I/O devices.

PIN CONFIGURATION

N, I PACKAGES



Pin. No.	Identifier	Function
1 – 8	UD7 – UD0	3-State, bidirectional User Data (UD) bus; UD0 and corresponds to $\overline{\text{IV0}}$.
9	$\overline{\text{UOC}}$	User Output Control — active low input to enable data output to UD0 – UD7.
10	$\overline{\text{UIC}}$	User Input Control — active low input to enable data input from UD0 – UD7.
11	$\overline{\text{ME}}$	Master Enable — active low input to enable the $\overline{\text{IV}}$ bus for data input, data output, or $\overline{\text{IV}}$ address selection/deselection: UD-bus operations are unaffected.
12	GND	Ground.
13	MCLK	Master Clock — active high input from Microcontroller used to strobe data into data latches from the $\overline{\text{IV}}$ bus and, for the synchronous 8X372, from the UD bus; MCLK also synchronizes $\overline{\text{IV}}$ address selection.
14	SC	Select Command — active high input from Microcontroller to enable $\overline{\text{IV}}$ address input from the $\overline{\text{IV}}$ bus for device selection.
15	WC	Write Command — active high input from Microcontroller to enable the writing of data into the data latches from the $\overline{\text{IV}}$ bus, provided $\overline{\text{UIC}}$ is not low.
16 – 23	$\overline{\text{IV0}}$ – $\overline{\text{IV7}}$	Interface Vector (Input/Output Bus) — 3-State, bidirectional, Microcontroller data bus; $\overline{\text{IV0}}$ corresponds to UD0.
24	V_{CC}	+5V power supply.

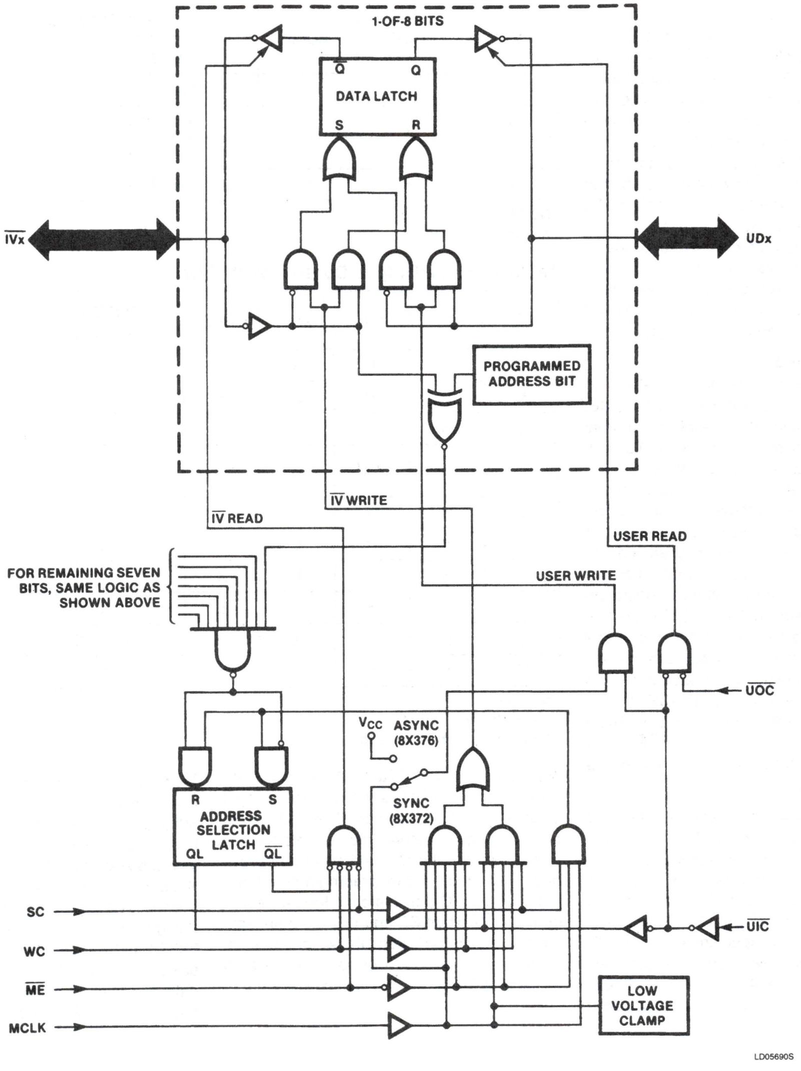

1-OF-8 BITS
$\overline{Q}$
Q
DATA LATCH
S
R
$\overline{IVx}$
UDx
PROGRAMMED ADDRESS BIT
$\overline{IV}$ READ
$\overline{IV}$ WRITE
USER READ
USER WRITE
FOR REMAINING SEVEN BITS, SAME LOGIC AS SHOWN ABOVE
$\overline{UOC}$
V_{CC}
ASYNC (8X376)
SYNC (8X372)
R
S
ADDRESS SELECTION LATCH
QL
$\overline{QL}$
SC
WC
$\overline{ME}$
MCLK
$\overline{UIC}$
LOW VOLTAGE CLAMP
LD05690S

Figure 1. Logic Diagram for 8X372/8X376 I/O Ports

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data (UD) bus interface is controlled by the $\overline{\mathrm{UIC}}$ and $\overline{\mathrm{UOC}}$ inputs. For the 8X372, data input from the UD bus is written synchronously with MCLK, that is, with $\overline{\mathrm{UIC}}$ low, information is written into the data latches only when MCLK is high. In the case of the 8X376, data input is asynchronous, in that, with $\overline{\mathrm{UIC}}$ low, data is latched in without regard to the level of MCLK. (Note: *To avoid the possibility of processor error when using the asynchronous 8X376, the $\overline{\mathrm{IV}}$ bus should not be read during the time the data latches are changing due to user input.*) Output drivers on the UD bus are enabled when UOC is low and $\overline{\mathrm{UIC}}$ is high.

$\overline{\mathrm{IV}}$ Bus Control

Input/output control of the $\overline{\mathrm{IV}}$ bus is shown in Table 2; this bus is controlled by SC, WC, $\overline{\mathrm{ME}}$, MCLK and the current state of the internal address selection latch. As shown in Table 2, $\overline{\mathrm{UIC}}$ is required to indicate priority of the UD bus for data input operations. The selection latch in the I/O port stores the result of the most recent $\overline{\mathrm{IV}}$ address selection. The latch is set when the internally preprogrammed address of the port matches the address on the $\overline{\mathrm{IV}}$ bus during an address-selection operation (SC = MCLK = High/WC = Low). The latch is cleared when the two 8-bit address patterns are in disagreement. The $\overline{\mathrm{IV}}$ bus can transfer data only when the selection latch is set. As shown in the APPLICATION DIAGRAM, the Microcontroller Left Bank ($\overline{\mathrm{LB}}$) and Right Bank ($\overline{\mathrm{RB}}$) outputs can control the $\overline{\mathrm{ME}}$ inputs for two banks of I/O devices, thus, acting as a ninth address bit.

Table 1. Input/Output Control of UD Bus

$\overline{\mathrm{UIC}}$	$\overline{\mathrm{UOC}}$	MCLK	FUNCTION OF UD BUS	
			8X372	8X376
H	L	X	Output data	Output data
L	X	H	Input data	Input data
L	X	L	Inactive	Input data
H	H	X	Inactive	Inactive

NOTE:
X = don't care

Table 2. Input/Output Control of IV Bus

$\overline{\mathrm{ME}}$	SC	WC	MCLK	$\overline{\mathrm{UIC}}$	SELECTION LATCH	FUNCTION OF $\overline{\mathrm{IV}}$ BUS
L	L	L	X	X	Set	Output Data
L	L	H	H	H	Set	Input Data
L	H	L	H	X	X	Input Address*
L	H	H	H	H	X	Input data and address*
L	H	H	H	L	X	Input Address*
L	X	H	L	X	X	Inactive
L	H	X	L	X	X	Inactive
L	L	H	H	L	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

NOTES:
X = don't care
* Selection latch is updated

Data is written into the data latches of a selected device from the $\overline{\mathrm{IV}}$ bus when WC, MCLK, and $\overline{\mathrm{UIC}}$ are all high and $\overline{\mathrm{ME}}$ is low. To prevent data-input conflicts, inputs from the $\overline{\mathrm{IV}}$ bus are inhibited when $\overline{\mathrm{UIC}}$ is low; under all other conditions, the $\overline{\mathrm{IV}}$ and UD busses operate independently. Output drivers on the $\overline{\mathrm{IV}}$ bus of a selected device are enabled when $\overline{\mathrm{ME}}$, WC, and SC are all low and the address selection latch is set. With SC and WC both high (shaded entry of Table 2), the bit pattern present on $\overline{\mathrm{IV0}}-\overline{\mathrm{IV7}}$ is interpreted as both input data and $\overline{\mathrm{IV}}$ address. Provided $\overline{\mathrm{UIC}}$ is high, the data is latched into the data latches whether or not the I/O port has been previously selected. If the preprogrammed address of the I/O port matches the bit pattern on $\overline{\mathrm{IV0}}-\overline{\mathrm{IV7}}$ when SC and WC are both high, the selection latch is set; otherwise, it is reset. (Note: *The Microcontroller never drives both SC and WC high at the same time.*)

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note: *A logic "1" in Microcontroller software corresponds to a high level on the UD bus even though the $\overline{\mathrm{IV}}$ bus is inverted.*) Both the 8X372 and 8X376 wakeup with the address selection latch in the unselected state and all data bits latched at the "logic 1" level (UD bus outputs high if enabled).

Addressable/Bidirectional I/O Ports | 8X372/8X376

ADDRESS PROGRAMMING AND ADDRESS PROTECT

Programming Procedures

Both 8X372 and 8X376 can be programmed to respond to any address within a range of 0_{10} through 255_{10}. In an unprogrammed state, low level ($\leqslant$ 0.8V) inputs on all $\overline{IV}$ bus lines (address 255_{10}) will select the device. To program a given address bit to match a high-level ($\geqslant$ 2V) input on the corresponding $\overline{IV}$ pin (a logical "0" to the Microcontroller), the counterpart UD-bus pin must be pulsed according to Table 3 and the following procedures:

Step 1: Set all control inputs to the inactive state — $\overline{UIC} = \overline{UOC} = \overline{ME} = V_{CC}$ and SC = WC = MCLK = GND; leave the UD and $\overline{IV}$ bus pins open.

Step 2: Increase V_{CC} to V_{CCP}.

Step 3: After V_{CC} has stabilized, apply a single programming pulse (Figure 2) to the user-bus bit that corresponds to the desired high-level $\overline{IV}$ address bit. The I/O port is programmed from the user bus (UD0 - UD7) for addressing from the Microcontroller bus ($\overline{IV0}$ - $\overline{IV7}$).

Step 4: Return V_{CC} to 0V. (Note: *If the programming of all address bits is completed in less than 1-second, V_{CC} can remain at 9V for the required interval of time.*)

Step 5: Steps 1 through 3 are applicable to the programming of each address bit that requires a high-level $\overline{IV}$ match.

Step 6: To verify that the address is properly programmed, return V_{CC} to +5V, set $\overline{IV0}$ - $\overline{IV7}$ to the desired (inverted) binary address pattern, set $\overline{ME}$ = WC = Low and SC = MCLK = High. If there are no programming errors, subsequent data written from $\overline{IV0}$ - $\overline{IV7}$ (WC = High) will appear inverted on UD0 - UD7.

Address Protect

After programming the I/O Port, steps should be taken to isolate the address circuits and make these circuits permanently immune to further change.

Table 3. Programming Specification

PARAMETERS	LIMITS Min	Typ	Max	UNITS
V_{CCP} — Programming supply voltage: Address	8.75	9	9.25	V
Protect		0		V
Maximum time $V_{CCP} > 5.25V$			1	Sec
Programming voltage: Address	8.75	9	9.25	V
Protect	8.75		9.25	V
Programming current: Address			5	mA
Protect			50	mA
t_r — Programming pulse rise time: Address	10		100	µS
Protect	10		100	µS
$t\omega$ — Programming pulse width	0.5		1	mS

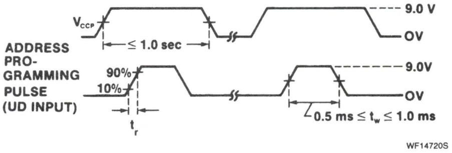


Figure 2. Address Programming Pulse

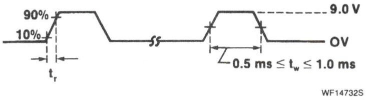


Figure 3. Protect Programming Pulse

Step 1: Set V_{CC} and all control inputs to 0V
$V_{CC} = \overline{UIC} = \overline{UOC} = \overline{ME}$ = SC = WC = MCLK = 0V); $\overline{IV0}$ - $\overline{IV7}$ = open circuit.

Step 2: Taking one pin at a time, apply a protect programming pulse (Figure 3) to each user-bus bit (UD0 - UD7) — refer to Table 3 for min/max specifications pertaining to voltage and current.

Step 3: Verify that the address circuits for each bit is isolated by applying 9V, in turn, to each user-bus pin (UD0 - UD7) and measuring less than 200 microamperes of input current. (Note: *Setup conditions are the same as those in Step 1.*)

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage[3]	+7	V_{DC}
V_I	Input voltage[3]	+5.5	V_{DC}
T_{STG}	Storage temperature range	-65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $4.75V \leqslant V_{CC} \leqslant 5.25V$, $0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{CC}	Supply voltage		4.75	5	5.25	V
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				0.8	V
V_{CL}	Input clamp voltage	V_{CC} = Min; I_I = -10mA			-1.5	V
I_{IH}	High-level input current[1]	V_{CC} = Max; V_{IH} = 2.7V		5	100	µA
I_{IL}	Low-level input current[1]	V_{CC} = Max; V_{IL} = 0.5V		-350	-550	µA
V_{OL}	Low-level output voltage $\overline{IV}$ bus ($\overline{IV0}$ – $\overline{IV7}$)	V_{CC} = Min; I_{OL} = 16mA			0.55	
	User bus (UD0 – UD7)	V_{CC} = Min; I_{OL} = 24mA			0.55	V
V_{OH}	High-level output voltage	V_{CC} = Min; I_{OH} = -3.2mA	2.4			V
I_{OS}	Short circuit output current[2] $\overline{IV}$ bus ($\overline{IV0}$ – $\overline{IV7}$)	V_{CC} = Max	-20			mA
	UD bus (UD0 – UD7)	V_{CC} = Max	-10			mA
I_{CC}	Supply current	V_{CC} = Max; $\overline{ME}$ = $\overline{UOC}$ = V_{CC}		90	150	mA

NOTES:

1. The input current includes the 3-State leakage current of the output driver on the data lines.
2. Only one output may be shorted at a time.
3. These limits do not apply during address programming.

ORDERING INFORMATION

DESCRIPTION	ORDERING CODE	
	Preprogrammed Addresses	Field Programmable Addresses
24-Pin plastic DIP	N8X372N-ZZZ	N8X372N or N8X376N
24-Pin ceramic DIP	N8X372I-ZZZ	N8X372I or N8X376I

NOTE:

Where, ZZZ = any address from 000_{10} through 255_{10}. I/O Ports with preprogrammed addresses 0_{10} through 15_{10} are stock items; consult the nearest Signetics Sales and Service Office when ordering parts in the address range of 16_{10} through 255_{10}.

AC ELECTRICAL CHARACTERISTICS $4.75V \leqslant V_{CC} \leqslant 5.25V$, $0°C \leqslant T_A \leqslant 70°C$

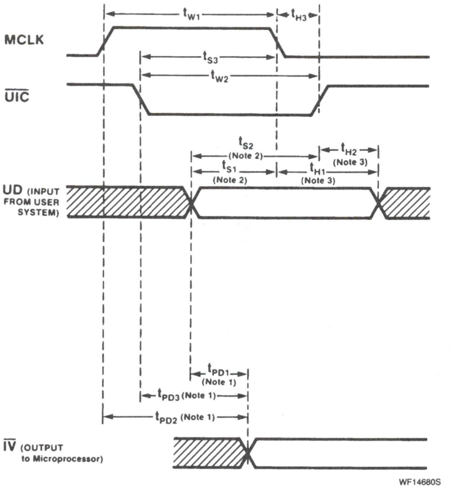
SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS[1]	LIMITS			UNIT
		From	To		Min	Typ	Max	
Pulse Widths:								
t_{W1}	Clock high	↑ MCLK	↓ MCLK		35			ns
t_{W2}	User input control	↓ $\overline{\text{UIC}}$	↑ $\overline{\text{UIC}}$	MCLK = High	35			ns
Propagation Delays:								
t_{PD1}	UD propagation delay	UD	$\overline{\text{IV}}$	MCLK = High SC = WC = $\overline{\text{ME}}$ = $\overline{\text{UIC}}$ = Low			30	ns
t_{PD2}	UD clock delay (8X732 only)	↑ MCLK	$\overline{\text{IV}}$	UD = Stable; SC = WC = $\overline{\text{ME}}$ = $\overline{\text{UIC}}$ = Low			50	ns
t_{PD3}	UD input delay	↓ $\overline{\text{UIC}}$	$\overline{\text{IV}}$	UD = Stable; MCLK = High; SC = WC = $\overline{\text{ME}}$ = Low			50	ns
t_{PD4}	$\overline{\text{IV}}$ data propagation delay	$\overline{\text{IV}}$	UD	MCLK = WC = $\overline{\text{UIC}}$ = High; $\overline{\text{ME}}$ = $\overline{\text{UOC}}$ = SC = Low			45	ns
t_{PD5}	$\overline{\text{IV}}$ data clock delay	↑ MCLK	UD	WC = $\overline{\text{UIC}}$ = High; $\overline{\text{IV}}$ = Stable, $\overline{\text{ME}}$ = $\overline{\text{UOC}}$ = SC = Low			55	ns
Output Enable Timing:								
t_{OE1}	UD output enable	↓ UOC	UD	$\overline{\text{UIC}}$ = High			30	ns
t_{OE2}	UD input recovery	↑ $\overline{\text{UIC}}$	UD	$\overline{\text{UOC}}$ = Low			30	ns
t_{OE3}	$\overline{\text{IV}}$ data master enable	↓ $\overline{\text{ME}}$	$\overline{\text{IV}}$	WC = SC = Low			22	ns
t_{OE5}	$\overline{\text{IV}}$ data write recovery	↓ WC	$\overline{\text{IV}}$	SC = $\overline{\text{ME}}$ = Low			25	ns
t_{OE6}	$\overline{\text{IV}}$ data select recovery	↓ SC	$\overline{\text{IV}}$	SC = $\overline{\text{ME}}$ = Low			25	ns
Output Disable Timing:								
t_{OD1}	UD output disable	↑ $\overline{\text{UOC}}$	UD	$\overline{\text{UIC}}$ = High			25	ns
t_{OD2}	UD input override	↓ $\overline{\text{UIC}}$	UD	$\overline{\text{UOC}}$ = Low			30	ns
t_{OD3}[2]	$\overline{\text{IV}}$ data master disable	↑ $\overline{\text{ME}}$	$\overline{\text{IV}}$	WC = SC = Low			20	ns
t_{OD4}[2]	$\overline{\text{IV}}$ data write override	↑ WC	$\overline{\text{IV}}$	SC = $\overline{\text{ME}}$ = Low			20	ns
t_{OD5}[2]	$\overline{\text{IV}}$ data select override	↑ SC	$\overline{\text{IV}}$	WC = $\overline{\text{ME}}$ = Low			20	ns

AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS[1]	LIMITS			UNIT
		From	To		Min	Typ	Max	
Setup Times:								
t_{S1}	UD clock setup time (8X372 only)	UD	↓ MCLK	$\overline{UIC}$ = Low	15			ns
t_{S2}	UD control setup time	UD	↑ $\overline{UIC}$	MCLK = High	15			ns
t_{S3}	User input control setup time (8X372 only)	↓ $\overline{UIC}$	↓ MCLK		25			ns
t_{S4}	$\overline{IV}$ data setup time	$\overline{IV}$	↓ MCLK	WC = High or SC = High; $\overline{ME}$ = Low; $\overline{UIC}$ = High	35			ns
t_{S5}[3]	$\overline{IV}$ master enable setup time	↓ ME	↓ MCLK	WC = High or SC = High; $\overline{UIC}$ = High	30			ns
t_{S6}	$\overline{IV}$ write control setup time	↑ WC	↓ MCLK	SC = $\overline{ME}$ = Low; $\overline{UIC}$ = High	30			ns
t_{S7}	$\overline{IV}$ select control setup time	↑ SC	↓ MCLK	WC = $\overline{ME}$ = Low	30			ns
Hold Times:								
t_{H1}	UD clock hold time (8X372 only)	↓ MCLK	UD	$\overline{UIC}$ = Low	15			ns
t_{H2}	UD control hold time	↑ $\overline{UIC}$	UD	MCLK = High	15			ns
t_{H3}	User input control hold time (8X372 only)	↓ MCLK	↑ $\overline{UIC}$		0			ns
t_{H4}	$\overline{IV}$ data hold time	↓ MCLK	$\overline{IV}$	WC = High or SC = High; $\overline{ME}$ = Low, $\overline{UIC}$ = High	5			ns
t_{H5}[3]	$\overline{IV}$ master enable hold time	↓ MCLK	↑ $\overline{ME}$	WC = High or SC = High; $\overline{UIC}$ = High	0			ns
t_{H6}	$\overline{IV}$ write control hold time	↓ MCLK	↓ WC	SC = $\overline{ME}$ = Low; $\overline{UIC}$ = High	0			ns
t_{H7}	$\overline{IV}$ select control hold time	↓ MCLK	↓ SC	WC = $\overline{ME}$ = Low	0			ns

NOTES:

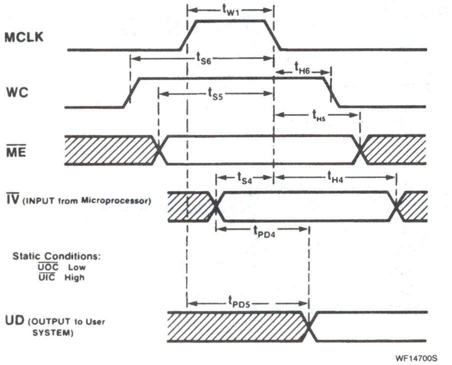
1. All measurements to the $\overline{IV}$ bus assumes the address selection latch is set.
2. These parameters are measured with a capacitive loading of 50pf and represent the output driver turn-off time.
3. If $\overline{ME}$ is to be High (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing into or selection of the I/O port.


MCLK
$\overline{\text{UIC}}$
UD (INPUT FROM USER SYSTEM)
$\overline{\text{IV}}$ (OUTPUT to Microprocessor)
t_{W1}
t_{H3}
t_{S3}
t_{W2}
t_{S2} (Note 2)
t_{H2} (Note 3)
t_{S1} (Note 2)
t_{H1} (Note 3)
t_{PD1} (Note 1)
t_{PD3} (Note 1)
t_{PD2} (Note 1)
WF14680S

STATIC CONDITIONS:
SC WC $\overline{\text{ME}}$ Low
$\overline{\text{UOC}}$ High

NOTES:
1. The actual time for stable data on the $\overline{\text{IV}}$ bus is the latest propagation from t_{PD1}, t_{PD2} and t_{PD3}.
2. The UD input must satisfy the setup-time requirements for both t_{S1} and t_{S2}.
3. Minimum hold-time required for the UD input is the earlier of the times specified by t_{H1} and t_{H2}.

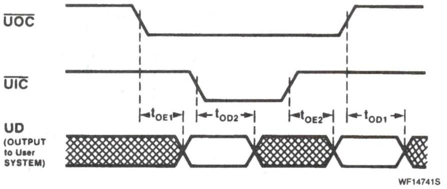
a. User Data Input Timing


MCLK
WC
$\overline{\text{ME}}$
$\overline{\text{IV}}$ (INPUT from Microprocessor)
Static Conditions:
$\overline{\text{UOC}}$ Low
$\overline{\text{UIC}}$ High
UD (OUTPUT to User SYSTEM)
t_{W1}
t_{S6}
t_{H6}
t_{S5}
t_{H5}
t_{S4}
t_{H4}
t_{PD4}
t_{PD5}
WF14700S

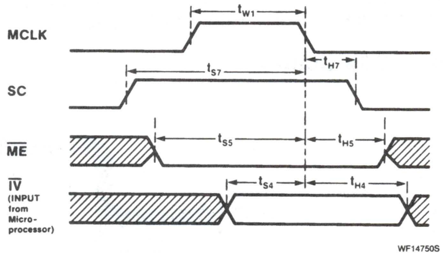
STATIC CONDITIONS:
$\overline{\text{UOC}}$ Low
$\overline{\text{UIC}}$ High

c. Microcontroller Write Cycle Timing

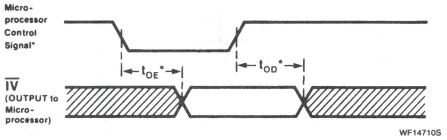
Legend:
▩ = 3-STATE
▨ = CHANGING DATA


$\overline{\text{UOC}}$
$\overline{\text{UIC}}$
UD (OUTPUT to User SYSTEM)
t_{OE1}
t_{OD2}
t_{OE2}
t_{OD1}
WF14741S

b. User Data Output Enable


MCLK
SC
$\overline{\text{ME}}$
$\overline{\text{IV}}$ (INPUT from Micro-processor)
t_{W1}
t_{H7}
t_{S7}
t_{S5}
t_{H5}
t_{S4}
t_{H4}
WF14750S

d. Microcontroller Select Cycle Timing

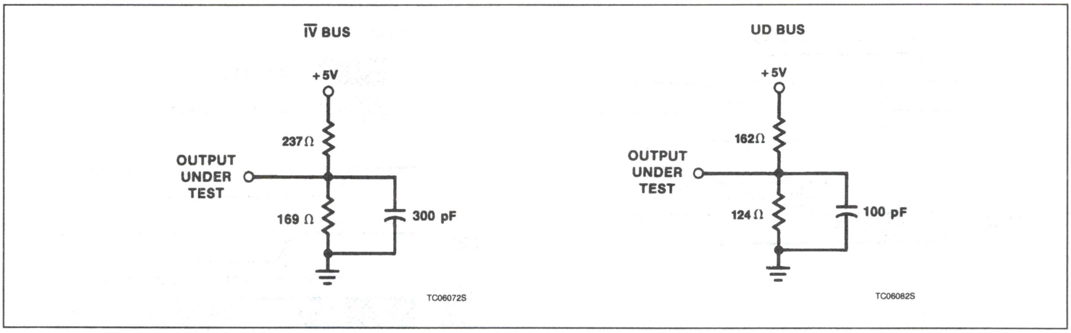

Micro-processor Control Signal*
$\overline{\text{IV}}$ (OUTPUT to Micro-processor)
t_{OE}*
t_{OD}*
WF14710S

***PARAMETER KEY**

MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS	STATIC CONDITIONS
$\overline{\text{ME}}$	t_{OE3} t_{OD3}	SC = WC = Low
WC	t_{OE4} t_{OD4}	SC = $\overline{\text{ME}}$ = Low
SC	t_{OE5} t_{OD5}	WC = $\overline{\text{ME}}$ = Low

e. Microcontroller Output Enable Timing

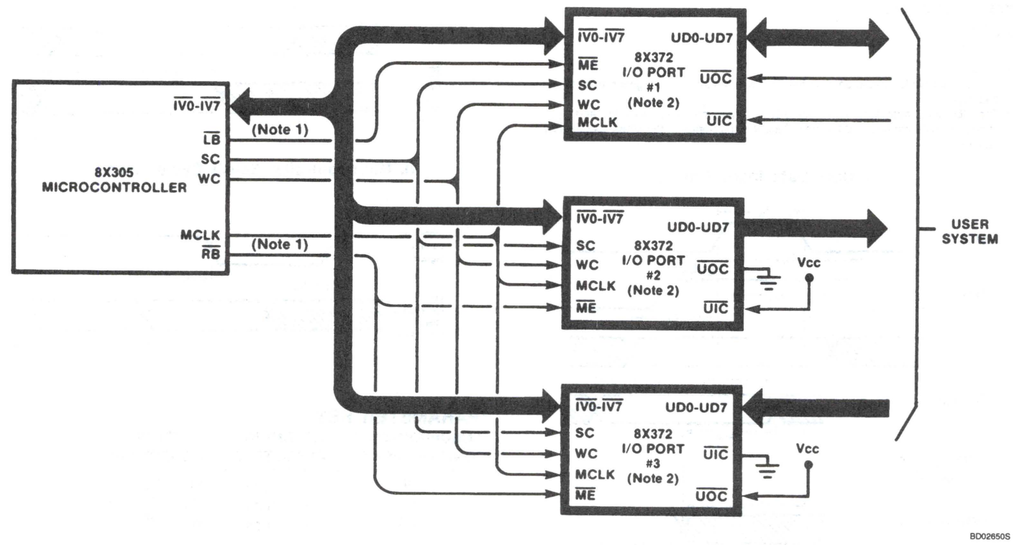
AC TEST LOADING CIRCUITS


$\overline{IV}$ BUS
+5V
237Ω
OUTPUT UNDER TEST
169 Ω
300 pF
TC06072S
UD BUS
+5V
162Ω
OUTPUT UNDER TEST
124 Ω
100 pF
TC06082S

APPLICATIONS

One way of using I/O Ports in a microprocessor-based system is shown in the following application diagram; there are many other ways of implementing I/O functions with these parts, both singly and in combination. By proper control the $\overline{UIC}$ and $\overline{UOC}$ lines, the user can implement bidirectional data transfers, exercise system control, and/or read system status. In the concept shown here, I/O Port #1 is setup for bidirectional data transfers and I/O Ports #2 and #3, respectively, serve as dedicated output and input devices.

APPLICATON DIAGRAM


8X305 MICROCONTROLLER
$\overline{IV0}$-$\overline{IV7}$
$\overline{LB}$
SC
WC
MCLK
$\overline{RB}$
(Note 1)
(Note 1)
$\overline{IV0}$-$\overline{IV7}$
UD0-UD7
$\overline{ME}$
SC
WC
MCLK
8X372 I/O PORT #1 (Note 2)
$\overline{UOC}$
$\overline{UIC}$
$\overline{IV0}$-$\overline{IV7}$
UD0-UD7
SC
WC
MCLK
$\overline{ME}$
8X372 I/O PORT #2 (Note 2)
$\overline{UOC}$
$\overline{UIC}$
Vcc
$\overline{IV0}$-$\overline{IV7}$
UD0-UD7
SC
WC
MCLK
$\overline{ME}$
8X372 I/O PORT #3 (Note 2)
$\overline{UIC}$
$\overline{UOC}$
Vcc
USER SYSTEM
BD02650S

NOTES:
1. Up to 256 discrete addresses can be assigned to each bank ($\overline{LB}$ or $\overline{RB}$).
2. The 8X376 can be implemented in an identical manner.

Signetics

8X374

Addressable/Bidirectional I/O Port With Parity

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics 8X374 is an addressable 8-bit I/O Port that features on-chip parity generate/check logic. The 8X374 port is designed for applications that require an 8-bit bidirectional interface element with parity-generate and parity-check capabilities. Typically, the 8X374 is used with the 8X305 Microcontroller and its associated Interface Vector (IV) bus.

As shown in the logic diagram of Figure 1, the 8X374 consists of eight identical latches, bits 0 through 7. These latches are accessed through either of two 8-bit busses, one connecting to the Microcontroller (IV bus) and the other to the user system (UD bus). Separate controls are provided for each bus and both busses operate independently, except when both attempt to input data at the same time. In such situations, the user bus always has priority. The data latches are transparent, in that, while either bus is enabled for input, all transitions in input data are propagated to the other bus, if enabled for output. The data latch in Figure 1 is common to both busses, that is, data traveling from the IV bus to the UD bus, or vice-versa, is latched and applied to the parity generate/check logic. The parity-bit latch is interfaced to the UD bus and latches the parity bit. The user can implement the parity features of the chip by simply selecting odd or even parity via the Parity SeLect (PSL) input pin. When data is output to the UD bus, a parity bit is generated and appended to each byte of data; for incoming data, parity is checked and the result is transmitted to an error-flag latch. The status of the latch (0 = no parity error/1 = parity error) is reflected by the Error Flag (EF) output pin. Operation of the error-flag latch is controlled by the Error Flag Hold (EFH) signal. With EFH low, the operation is transparent; when high the contents of the latch are frozen to avoid false errors while data latches are changing.

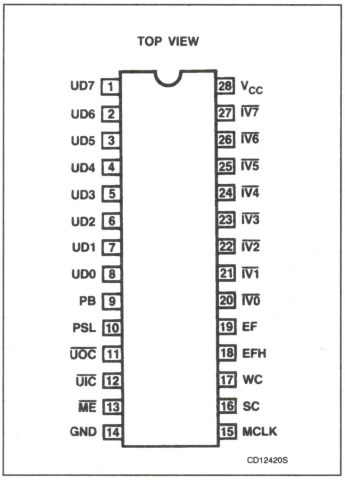
The 8X374 is available with either preprogrammed addresses (0_{10} to 255_{10}) or unprogrammed; the device can be field-programmed over the same address range as the preprogrammed port. Input/Output operations to the Microcontroller bus can begin once the 8X374 enabling address has been selected and appropriate control signals from the IV bus are generated. Port selection is implemented by putting the 8X374 address (0_{10} to 255_{10}) on the IV bus. Once selected, the I/O port remains selected until a different port address is put on the bus.

With appropriate control inputs, data is accessible on the UD bus at all times. A Master Enable (ME) input, which is typically connected to the Left Bank (LB) or Right Bank (RB) output of the Microcontroller, provides the capability of organizing the IV bus into two separate and independent banks of I/O devices.

FEATURES

- **Two bidirectional 8-bit busses**
- **Independent bus operation (user bus priority for data entry)**
- **Parity generate/check logic with:**
  - **Odd/Even parity select**
  - **Strobed error flag output**
- **Synchronous data input**
- **Programmable Microcontroller port address**
- **3-State TTL outputs (for all except parity error flag)**
- **High drive capabilities**
- **Power-up to predetermined state**
- **Directly compatible with 8X305 Microcontroller**
- **Single +5V supply**
- **0.6 inch, 28-pin DIP**

PIN CONFIGURATION



Addressable/Bidirectional I/O Port With Parity 8X374

ORDERING INFORMATION

DESCRIPTION	ORDERING CODE	
	Preprogrammed Addressed	**Field Programmable Addresses**
28-Pin plastic DIP	N8X374N-ZZZ	N8X374N
28-Pin ceramic DIP	N8X374F-ZZZ	N8X374F

NOTE:
Where zzz = any address from 000_{10} through 255_{10}. I/O Ports with preprogrammed addresses 0_{10} through 15_{10} are stock items; consult the nearest Signetics Sales and Service Office when ordering parts in the address range of 16_{10} through 255_{10}.

PIN DESCRIPTION

PIN NO	IDENTIFIER	FUNCTION
1 – 8	UD7 – UD0	3-State bidirectional **User Data** (UD) bus; **UD0** corresponds to $\overline{\text{IV0}}$.
9	PB	User port **Parity Bit** I/O pin.
10	PSL	**Parity SeLect** input control; even parity = 1 and odd parity = 0.
11	$\overline{\text{UOC}}$	**User Output Control**; Active low input to enable data output from UD0 – UD7.
12	$\overline{\text{UIC}}$	**User Input Control**: Active low input to enable data input from UD0 – UD7.
13	$\overline{\text{ME}}$	**Master Enable**: Active low input to enable the IV bus from data input, data output, or IV address selection/deselection; UD-bus operations are unaffected.
14	GND	**GrouND**
15	MCLK	**Master Clock**: Active high input from Microcontroller used to strobe data into the data latches; MCLK also synchronizes IV address selection.

Addressable/Bidirectional I/O Port With Parity 8X374

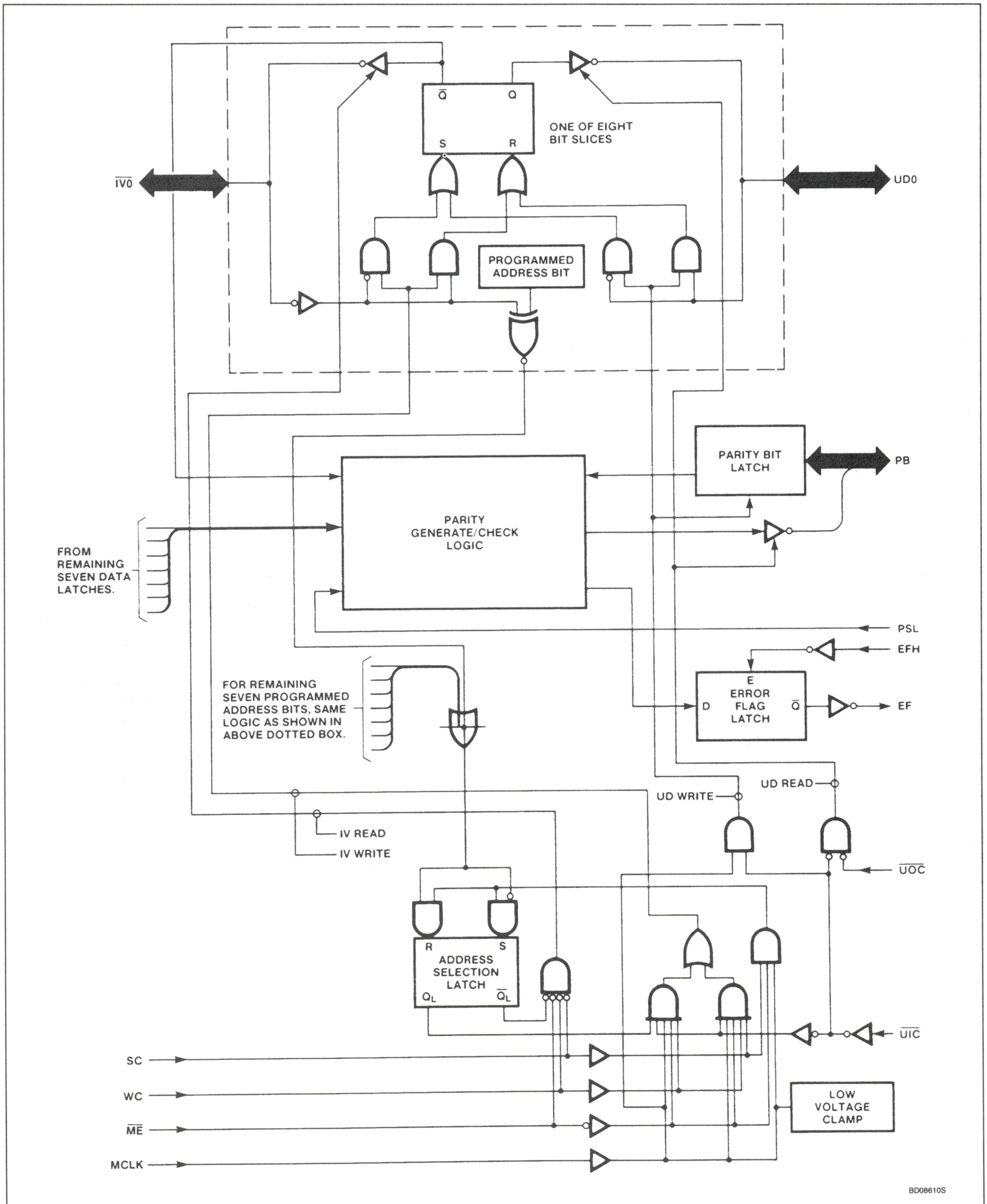

Q̄
Q
S
R
ONE OF EIGHT
BIT SLICES
IV̄0
UD0
PROGRAMMED ADDRESS BIT
PARITY BIT LATCH
PB
PARITY GENERATE/CHECK LOGIC
FROM REMAINING SEVEN DATA LATCHES.
PSL
EFH
E
ERROR FLAG LATCH
D
Q̄
EF
FOR REMAINING SEVEN PROGRAMMED ADDRESS BITS, SAME LOGIC AS SHOWN IN ABOVE DOTTED BOX.
UD READ
UD WRITE
IV READ
IV WRITE
UŌC
R
S
ADDRESS SELECTION LATCH
Q_L
$\overline{Q}_L$
UĪC
SC
WC
M̄Ē
MCLK
LOW VOLTAGE CLAMP
BD08610S

Figure 1. Logic Diagram for 8X374 I/O Port

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data (UD) bus and parity-bit interface are controlled by the UIC and UOC inputs. Data from the UD bus is written synchronously with MCLK, that is with UIC low, information is written into the data latches only when MCLK is high. Output drivers on the UD bus are enabled when UOC is low and UIC is high.

Table 1. Input/Output Control of UD Bus

UIC	UOC	MCLK	FUNCTION OF UD BUS	
			8-Bit Data Bus	Parity Bit
H	L	X	Output data	Output parity
L	X	H	Input data	Input parity
L	X	L	Inactive	Inactive
H	H	X	Inactive	Inactive

NOTE:
X = Don't Care.

IV Bus Control

Input/Output control of the IV bus is shown in Table 2; this bus is controlled by SC, WC, ME, MCLK and the current state of the internal address selection latch. As shown in Table 2, UIC is required to indicate priority of the UD bus for data input operations. The selection latch in the I/O port stores the result of the most recent IV address selection. The latch is set when the internally preprogrammed address of the port matches the address on the IV bus during an address-selection operation (SC = MCLK = High; ME = WC = Low). The latch is cleared when the two 8-bit address patterns are in disagreement. The IV bus can transfer data only when the selection latch is set. As shown in the Application Diagram, the 8X305 Left Bank (LB) and Right Bank (RB) outputs can control the ME inputs for two banks of I/O devices, thus, acting as a ninth address bit.

Data is written into the data latches of a selected device from the IV bus when WC, MCLK and UIC are all high and ME is low. To prevent data-input conflicts, inputs from the IV bus are inhibited when UIC is low; under all other conditions, the IV and UD busses operate independently. Output drivers on the IV bus of a selected device are enabled when ME, WC, and SC are all low and the address selection latch is set.

Table 2. Input/Output Control of IV Bus

ME	SC	WC	MCLK	UIC	SELECTION LATCH	FUNCTION OF IV BUS
L	L	L	X	X	Set	Output data
L	L	H	H	H	Set	Input data
L	H	L	H	X	X	Input address*
L	X	H	L	X	X	Inactive
L	H	X	L	X	X	Inactive
L	L	H	H	L	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

NOTE:
X = Don't Care.
*Selection latch is updated.

Parity Generate/Check Logic

The Parity Bit (PB) pin provides both parity-generate and parity-check capabilities, according to user data bus controls. With UIC low (active), a parity check is performed on the input data stream; with UOC low (active) and UIC high, the 8X374 generates the parity-bit for the output data stream. The user can select odd or even parity via the Parity SeLect (PSL) input control, 1 = even parity and 0 = odd parity. As data and parity are input to the data latches and the parity-bit latch from the UD bus and PB line (Figure 1), parity errors (if any) are continuously detected by the parity-check logic. Parity error status enters the error flag latch (if enabled) and appears at the EF output pin. The error latch can be strobed by the Error Flag Hold (EFH) control to latch in valid error status; otherwise, the error flag is transparent to the user. (Note: If the system uses less than eight data bits, keeping zeros in unused data latches preserves proper parity operation.)

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note: A logic "1" in Microcontroller software corresponds to a high-level on the UD bus even though the IV bus is inverted.) The 8X374 wakes up with the address selection latch in the unselected state, all data bits latched at the "logic 1" level (UD bus outputs high if enabled), and the EF output high.

ADDRESS PROGRAMMING AND ADDRESS PROTECT

Programming Procedures

The 8X374 can be programmed to respond to any address within a range of 0_{10} through 255_{10}. In an unprogrammed state, low level ($\leqslant$0.8V) inputs on all IV bus lines (address 255_{10}) will select the device. To program a given address bit to match a high-level ($\geqslant$2.0V) input on the corresponding IV pin (a logical "0" to the Microcontroller), the counterpart UD-bus pin must be pulsed according to Table 3 and the following procedures:

Step 1: Set all control inputs to the inactive state, UIC = UOC = ME = V_{CC} and SC = WC = MCLK = 0V; leave the UD and IV bus pins open.

Step 2: Increase V_{CC} to V_{CCP}.

Step 3: After V_{CC} has stabilized, apply a single programming pulse (Figure 2) to the user-bus bit that corresponds to the desired high-level IV address bit. The I/O port is programmed from the user bus (UD0-UD7) for addressing from the Microcontroller bus (IV0-IV7).

Step 4: Return V_{CC} to 0 volts. (Note: If the programming of all address bits is completed in less than one second, V_{CC} can remain at V_{CCP} for the required interval of time.)

Step 5: Step 1 through Step 3 are applicable to the programming of each address bit that requires a high-level IV match.

Step 6: To verify that the address is properly programmed, return V_{CC} to +5V and set IV0-IV7 to the desired address pattern (inverted). Set ME = WC = Low and SC = MCLK = High to select the programmed I/O port. With ME = SC = Low and WC = MCLK = High, write an 8-bit pattern to the port. If there are no programming errors, the transmitted data pattern will appear inverted at UD0-UD7 of selected port.

Table 3. Programming Specifications

PARAMETERS	LIMITS			UNITS
	Min	Typ	Max	
V_{CCP} — Programming supply voltage:				
Address	8.75	9.0	9.25	V
Protect		0		V
Maximum Time V_{CC} > 5.25V			1.0	sec
Programming voltage:				
Address	8.75	9.0	9.25	V
Protect	8.75		9.25	V
Programming current:				
Address			5	mA
Protect			50	mA
t_r — Programming pulse rise time:				
Address	10		100	μs
Protect	10		100	μs
t_w — Programming pulse width				

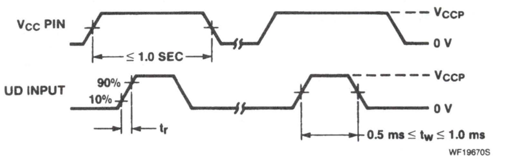


Figure 2. Address Programming Pulse

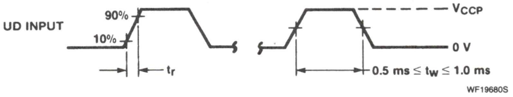


Figure 3. Protect Programming Pulse

ADDRESS PROTECT

After programming the I/O Port, optional steps can be taken to isolate the fuse circuits and to make these circuits permanently immune to further change.

Step 1: Set V_{CC} and all control inputs to 0 volts;
V_{CC} = UIC = UOC = ME = SC = WC = MCLK = 0V,
IV0-IV7 = open circuit.

Step 2: Taking one pin at a time, apply a protect programming pulse (Figure 3) to each user-bus bit (UD0-UD7). Refer to Table 3 for min/max specifications pertaining to voltage and current.

Step 3: Verify that the address circuits for each bit are isolated by applying V_{CCP} to each user-bus pin (UD0-UD7), and measuring less than 200mA of input current. (Note: Setup conditions are the same as those in Step 1.)

Addressable/Bidirectional I/O Port With Parity 8X374

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage[3]	+7	V_{DC}
V_{IN}	Input voltage[3]	+5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V (\pm 5\%)$; $T_A \geqslant 0°C$, $T_A \leqslant 70°C$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
V_{CC}	Supply Voltage		4.75	5	5.25	V
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				0.8	V
V_{CL}	Input clamp voltage	V_{CC} = Min; I_I = −10mA			−1.5	V
I_{IH}	High-level input current[1]	V_{CC} = Max; V_{IH} = 2.7V		5	100	μA
I_{IL}	Low-level input current[1]	V_{CC} = Max; V_{IL} = 0.5V		−350	−550	μA
V_{OL}	Low-level output voltage $\overline{IV}$ bus ($\overline{IV0}$ – $\overline{IV7}$)	V_{CC} = Min; I_{OL} = 16mA			0.55	V
	User bus (UD0 – UD7) and PB	V_{CC} = Min; I_{OL} = 24mA			0.55	V
	EF	V_{CC} = Min; I_{OL} = 8mA			0.55	V
V_{OH}	High-level output voltage EF	V_{CC} = Min; I_{OH} = −1mA	2.4			V
	Others	V_{CC} = Min; I_{OH} = −3.2mA	2.4			V
I_{OS}	Short circuit output current[2] $\overline{IV}$ bus ($\overline{IV0}$ – $\overline{IV7}$)	V_{CC} = Max	−20			mA
	UD bus (UD0 – UD7)	V_{CC} = Max	−10			mA
I_{CC}	Supply current	V_{CC} = Max; $\overline{ME}$ = $\overline{UOC}$ = V_{CC}		90	150	mA

NOTES:
1. The input current includes the high-Z leakage current of the output drivers ($\overline{IV0}$ – $\overline{IV7}$, UD0 – UD7) on the data lines.
2. Only one output may be shorted at a time for testing purposes.
3. These limits do not apply during address programming.

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5V$ (±5%); $T_A \geqslant 0°C$, $T_A \leqslant 70°C$

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS[1]	LIMITS			UNIT
		From	To		Min	Typ	Max	
Pulse width								
t_{W1}	Clock high	↑ MCLK	↓ MCLK		35			ns
t_{W2}	User input control	↓ $\overline{\text{UIC}}$	↑ $\overline{\text{UIC}}$	MCLK = High	35			ns
Propagation delay								
t_{PD1}	UD propagation delay	UD	IV	MCLK = High SC = WC = $\overline{\text{ME}}$ = $\overline{\text{UIC}}$ = Low			40	ns
t_{PD2}	UD clock delay	↑ MCLK	$\overline{\text{IV}}$	UD = Stable; SC = WC = $\overline{\text{ME}}$ = $\overline{\text{UIC}}$ = Low			50	ns
t_{PD3}	UD input delay	↓ $\overline{\text{UIC}}$	$\overline{\text{IV}}$	UD = Stable; MCLK = High; SC = WC = $\overline{\text{ME}}$ = Low			50	ns
t_{PD4}	$\overline{\text{IV}}$ data propagation delay	$\overline{\text{IV}}$	UD	MCLK = WC = $\overline{\text{UIC}}$ = High; $\overline{\text{ME}}$ = $\overline{\text{UOC}}$ = SC = Low			45	ns
t_{PD5}	$\overline{\text{IV}}$ data clock delay	↑ MCLK	UD	WC = $\overline{\text{UIC}}$ = High; $\overline{\text{IV}}$ = Stable, $\overline{\text{ME}}$ = $\overline{\text{UOC}}$ = SC = Low			55	ns
t_{PD6}	Error flag propagation delay	UD, PB	EF	MCLK = High; $\overline{\text{UIC}}$ = EFH = Low			55	ns
t_{PD7}	Parity generate propagation delay	$\overline{\text{IV}}$	PB	MCLK = WC = $\overline{\text{UIC}}$ = High; $\overline{\text{UOC}}$ = $\overline{\text{ME}}$ = Low			55	ns
t_{PD8}	Error flag strobe delay[3]	↓ EFH	EF				20	ns
Output enable timing								
t_{OE1}	UD output enable	↓ $\overline{\text{UOC}}$	UD, PB	$\overline{\text{UIC}}$ = High			30	ns
t_{OE2}	UD input recovery	↑ $\overline{\text{UIC}}$	UD, PB	$\overline{\text{UOC}}$ = Low			30	ns
t_{OE3}	$\overline{\text{IV}}$ data master enable	↓ $\overline{\text{ME}}$	$\overline{\text{IV}}$	WC = SC = Low			22	ns
t_{OE4}	$\overline{\text{IV}}$ data write recovery	↓ WC	$\overline{\text{IV}}$	SC = $\overline{\text{ME}}$ = Low			25	ns
t_{OE5}	$\overline{\text{IV}}$ data select recovery	↓ SC	$\overline{\text{IV}}$	SC = $\overline{\text{ME}}$ = Low			25	ns
Output disable timing								
t_{OD1}	UD output disable	↑ $\overline{\text{UOC}}$	UD, PB	$\overline{\text{UIC}}$ = High			25	ns
t_{OD2}	UD input override	↓ $\overline{\text{UIC}}$	UD, PB	$\overline{\text{UOC}}$ = Low			30	ns
t_{OD3}[4]	$\overline{\text{IV}}$ data master disable	↑ $\overline{\text{ME}}$	$\overline{\text{IV}}$	WC = SC = Low			20	ns
t_{OD4}[4]	$\overline{\text{IV}}$ data write override	↑ WC	$\overline{\text{IV}}$	SC = $\overline{\text{ME}}$ = Low			20	ns
t_{OD5}[4]	$\overline{\text{IV}}$ data select override	↑ SC	$\overline{\text{IV}}$	WC = $\overline{\text{ME}}$ = Low			20	ns

Addressable/Bidirectional I/O Port With Parity 8X374

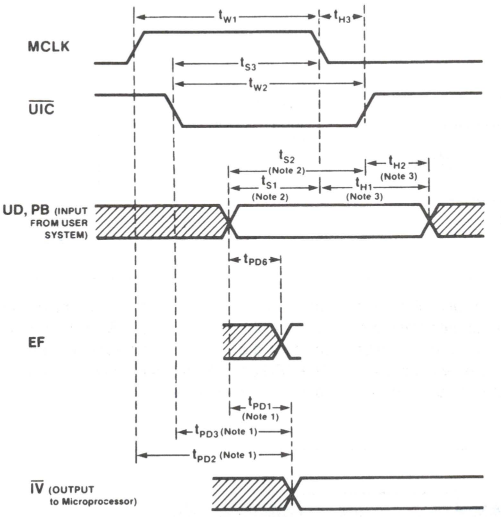
AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS[1]	LIMITS			UNIT
		From	To		Min	Typ	Max	
Setup times								
t_{S1}	UD clock setup time	UD, PB	↓ MCLK	$\overline{\text{UIC}}$ = Low	15			ns
t_{S2}	UD control setup time	UD, PB	↑ $\overline{\text{UIC}}$	MCLK = High	15			ns
t_{S3}	User input control setup time	↓ $\overline{\text{UIC}}$	↓ MCLK		25			ns
t_{S4}	$\overline{\text{IV}}$ data setup time	$\overline{\text{IV}}$	↓ MCLK	WC = High or SC = High; $\overline{\text{ME}}$ = Low; $\overline{\text{UIC}}$ = High	35			ns
t_{S5}[2]	$\overline{\text{IV}}$ master enable setup time	↓ $\overline{\text{ME}}$	↓ MCLK	WC = High or SC = High; $\overline{\text{UIC}}$ = High	30			ns
t_{S6}	$\overline{\text{IV}}$ write control setup time	↑ WC	↓ MCLK	SC = $\overline{\text{ME}}$ = Low; $\overline{\text{UIC}}$ = High	30			ns
t_{S7}	$\overline{\text{IV}}$ select control setup time	↑ SC	↓ MCLK	WC = $\overline{\text{ME}}$ = Low	30			ns
Hold times								
t_{H1}	UD clock hold time	↓ MCLK	UD, PB	$\overline{\text{UIC}}$ = Low	15			ns
t_{H2}	UD control hold time	↑ $\overline{\text{UIC}}$	UD, PB	MCLK = High	15			ns
t_{H3}	User input control hold time	↓ MCLK	↑ $\overline{\text{UIC}}$		0			ns
t_{H4}	$\overline{\text{IV}}$ data hold time	↓ MCLK	$\overline{\text{IV}}$	WC = High or SC = High; $\overline{\text{ME}}$ = Low, $\overline{\text{UIC}}$ = High	5			ns
t_{H5}[2]	$\overline{\text{IV}}$ master enable hold time	↓ MCLK	↑ $\overline{\text{ME}}$	WC = High or SC = High; $\overline{\text{UIC}}$ = High	0			ns
t_{H6}	$\overline{\text{IV}}$ write control hold time	↓ MCLK	↓ WC	SC = $\overline{\text{ME}}$ = Low; $\overline{\text{UIC}}$ = High	0			ns
t_{H7}	$\overline{\text{IV}}$ select control hold time	↓ MCLK	↓ SC	WC = $\overline{\text{ME}}$ = Low	0			ns

NOTES:

1. All measurements to the $\overline{\text{IV}}$ bus assumes the address selection latch is set.
2. If $\overline{\text{ME}}$ is to be high (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing into or selection of the I/O port.
3. Parameters are measured by holding $\overline{\text{UIC}}$ = High and MCLK = Low and changing the state of the PSL input before each EFH pulse.
4. There parameters are measured with a capacitive loading of 50pF and represent the output driver turn-off time.

TIMING DIAGRAMS



t_{W1}
t_{H3}
MCLK
t_{S3}
t_{W2}
$\overline{UIC}$
t_{S2} (Note 2)
t_{H2} (Note 3)
t_{S1} (Note 2)
t_{H1} (Note 3)
UD, PB (INPUT FROM USER SYSTEM)
t_{PD6}
EF
t_{PD1} (Note 1)
t_{PD3} (Note 1)
t_{PD2} (Note 1)
$\overline{IV}$ (OUTPUT to Microprocessor)

WF19690S

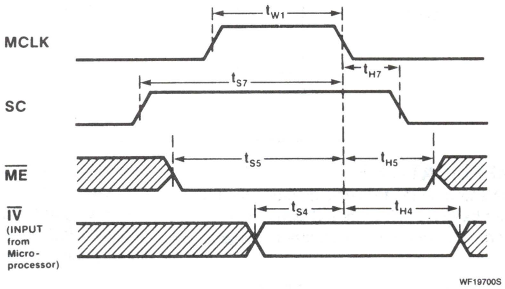
STATIC CONDITIONS:
SC = WC = $\overline{ME}$ = EFH = Low; $\overline{UOC}$ High

Notes:

1. The actual time for stable data on the $\overline{IV}$ bus is the latest propagation from t_{PD1}, t_{PD2}, and t_{PD3}.
2. The UD input must satisfy the setup-time requirements for both t_{S1} and t_{S2}.
3. Minimum hold-time required for the UD input is the earlier of the times specified by t_{H1} and t_{H2}.

User Data Input Timing

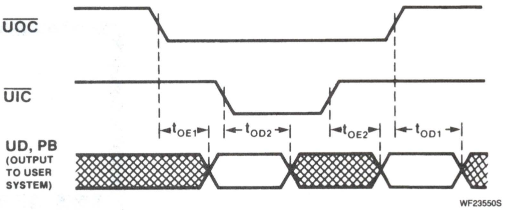
Legend:
CHANGING DATA ▨
Hi-Z ▩
WF14600S



t_{W1}
MCLK
t_{H7}
t_{S7}
SC
t_{S5}
t_{H5}
$\overline{ME}$
t_{S4}
t_{H4}
$\overline{IV}$ (INPUT from Micro-processor)

WF19700S

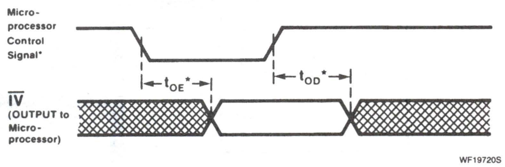
Microcontroller



$\overline{UOC}$
$\overline{UIC}$
t_{OE1}
t_{OD2}
t_{OE2}
t_{OD1}
UD, PB (OUTPUT TO USER SYSTEM)

WF23550S

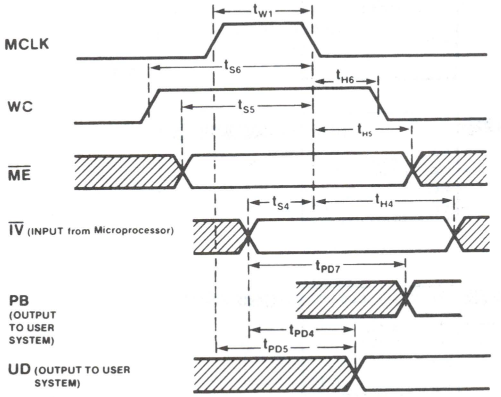
User Data Output Enable Timing



Micro-processor Control Signal*
t_{OE}*
t_{OD}*
$\overline{IV}$ (OUTPUT to Micro-processor)

WF19720S

Microcontroller Output Enable Timing



t_{W1}
MCLK
t_{S6}
t_{H6}
WC
t_{S5}
t_{H5}
$\overline{ME}$
t_{S4}
t_{H4}
$\overline{IV}$ (INPUT from Microprocessor)
t_{PD7}
PB (OUTPUT TO USER SYSTEM)
t_{PD4}
t_{PD5}
UD (OUTPUT TO USER SYSTEM)

Static Conditions:
$\overline{UOC}$ - Low
$\overline{UIC}$ - High

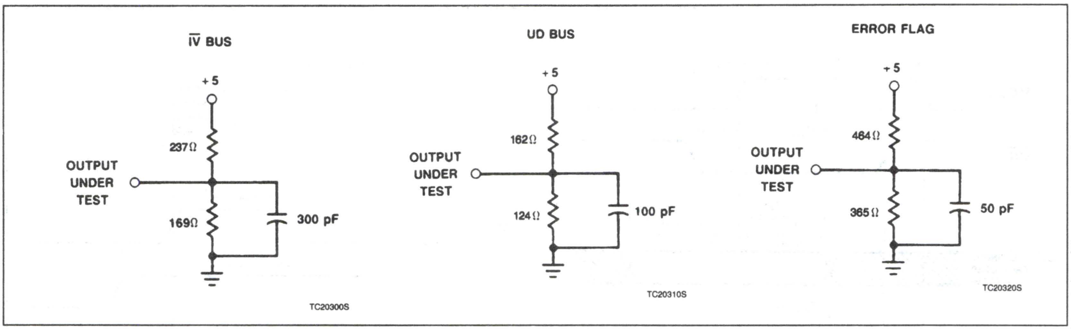
WF19710S

Microcontroller Write Cycle Timing

*PARAMETER KEY

MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS	STATIC CONDITIONS
$\overline{ME}$	t_{OE3} t_{OD3}	SC = WC = LOW
WC	t_{OE4} t_{OD4}	SC = $\overline{ME}$ = LOW
SC	t_{OE5} t_{OD5}	WC = $\overline{ME}$ = LOW

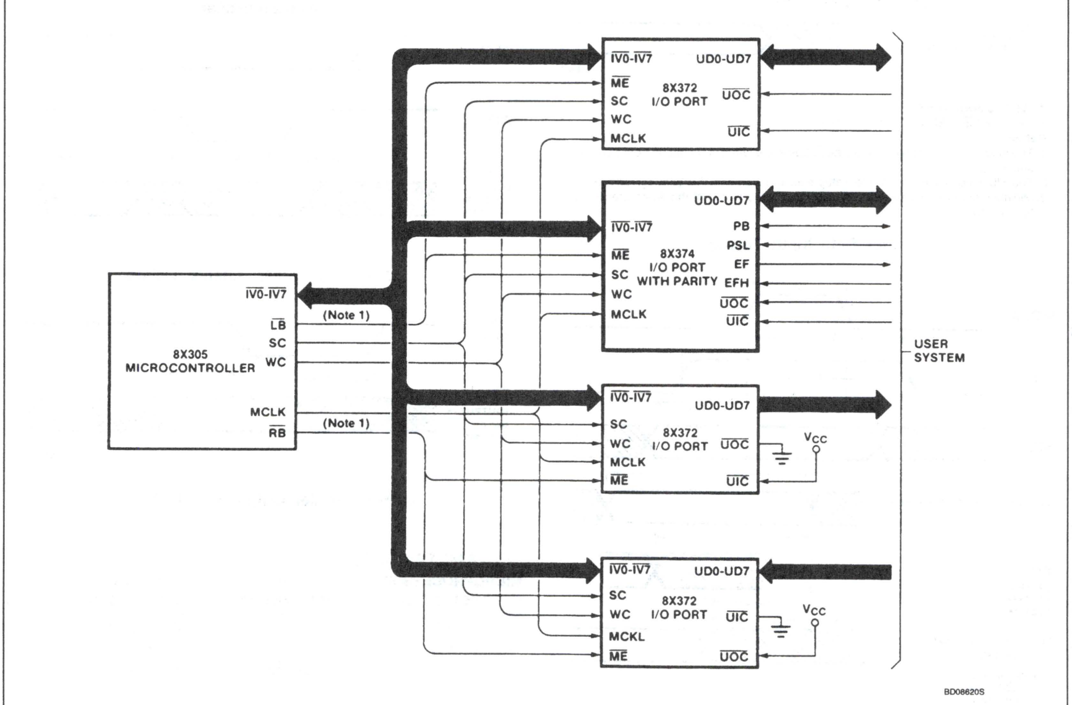
TEST LOADING CIRCUITS


$\overline{\text{IV}}$ BUS
+5
237Ω
OUTPUT UNDER TEST
169Ω
300 pF
TC20300S
UD BUS
+5
162Ω
OUTPUT UNDER TEST
124Ω
100 pF
TC20310S
ERROR FLAG
+5
464Ω
OUTPUT UNDER TEST
365Ω
50 pF
TC20320S

APPLICATIONS

As shown in the following diagram, the 8X374 can be used with other I/O ports to provide a complete range of input/output functions. By proper control of the UIC and UOC lines, the user can perform bidirectional data transfers, exercise system control, read system status and, by using the 8X374, implement a bidirectional parity-controlled data stream. To use the parity capabilities, the user need only select even or odd parity (PSL = 1 or 0) and connect the PB pin to the system parity bit. The EFH and EF pins can be wired according to system requirements.

APPLICATIONS DIAGRAM


8X305 MICROCONTROLLER
$\overline{\text{IV0}}$-$\overline{\text{IV7}}$
$\overline{\text{LB}}$
SC
WC
MCLK
$\overline{\text{RB}}$
(Note 1)
(Note 1)
$\overline{\text{IV0}}$-$\overline{\text{IV7}}$
UD0-UD7
$\overline{\text{ME}}$
SC
WC
MCLK
8X372 I/O PORT
$\overline{\text{UOC}}$
$\overline{\text{UIC}}$
UD0-UD7
$\overline{\text{IV0}}$-$\overline{\text{IV7}}$
PB
PSL
EF
EFH
$\overline{\text{UOC}}$
$\overline{\text{UIC}}$
$\overline{\text{ME}}$
SC
WC
MCLK
8X374 I/O PORT WITH PARITY
$\overline{\text{IV0}}$-$\overline{\text{IV7}}$
UD0-UD7
SC
WC
MCLK
$\overline{\text{ME}}$
8X372 I/O PORT
$\overline{\text{UOC}}$
$\overline{\text{UIC}}$
V_{CC}
$\overline{\text{IV0}}$-$\overline{\text{IV7}}$
UD0-UD7
SC
WC
MCKL
$\overline{\text{ME}}$
8X372 I/O PORT
$\overline{\text{UIC}}$
$\overline{\text{UOC}}$
V_{CC}
USER SYSTEM
BD08620S

NOTE:
1. Up to 256 discrete addresses can be assigned to each bank ($\overline{\text{LB}}$ or $\overline{\text{RB}}$).

Signetics

Microprocessor Products

8X382

4-Input/4-Output Addressable I/O Port

Product Specification

DESCRIPTION

The 8X382 I/O Port is an addressable, 3-State device designed for use as an interface element in systems that use TTL-compatible busses. Typically, the 8X382 is used with the 8X305 Microcontroller and its associated Interface Vector ($\overline{IV}$) bus; however, it can also be used with the 8X300 Microcontroller or an equivalent microprocessor. The 8X382 is functionally the same and pin-for-pin compatible with the older 8X42; however, the new port features better performance, increased drive current, and improved programming procedures.

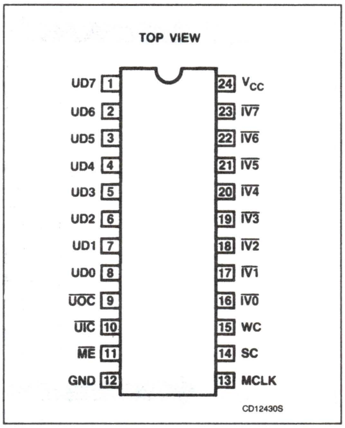
As shown in the logic diagram of Figure 1, the I/O port consists of eight data latches — bits 0 through 7. These latches are accessed through either of two busses — an 8-bit bidirectional $\overline{IV}$ bus connected to the Microcontroller and a User Data (UD) bus consisting of four dedicated inputs (bits UD0 through UD3) and four dedicated outputs (bits UD4 through UD7). All eight bits may be read from or four data bits ($\overline{IV4}$ – $\overline{IV7}$) can be written into via the $\overline{IV}$ bus; eight bits of I/O address can be written from the $\overline{IV}$ bus. Separate controls are provided for each bus and both busses operate independently. The I/O data latches are transparent, in that, when either bus is enabled for input, all transitions in input data are propagated to the other bus, if that bus in enabled, for output.

The 8X382 is available with preprogrammed addresses (0_{10} through 255_{10}); it can also be field-programmed over the same address range. Input/Output operations can begin once the I/O port is selected and appropriate control signals are generated. Port selection is implemented by putting the I/O port address (0_{10} – 255_{10}) on the $\overline{IV}$ bus; once selected, the I/O port remains selected until a different "port address" is put on the bus. Thus, software overhead is minimized. Data is accessible on the UD bus at all times. A Master Enable ($\overline{ME}$) input, which is typically connected to the Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) output of the Microcontroller, provides the capability of organizing the $\overline{IV}$ bus into two separate and independent banks of I/O devices.

FEATURES

- **Bidirectional 8-bit Microcontroller ($\overline{IV}$) bus**
- **User bus — four input bits and four output bits**
- **Independent bus operation**
- **Synchronous user data input**
- **Programmed Microcontroller port address**
- **3-State TTL outputs with high-drive capabilities**
- **Power-up to predetermined logic state**
- **Directly compatible with 8X305 or 8X300 Microcontrollers**
- **Single +5V supply**
- **0.4 inch 24-pin DIP**

PIN CONFIGURATION



PIN DESCRIPTION

PIN NO	IDENTIFIER	FUNCTION
1 – 4	UD7 – UD4	3-State, dedicated output lines for user data; UD7 corresponds to $\overline{IV7}$.
5 – 8	UD3 – UD0	Dedicated input lines for user data; UD0 corresponds to $\overline{IV0}$.
9	$\overline{UOC}$	User Output Control — active low input to enable data output to UD4 – UD7.
10	$\overline{UIC}$	User Input Control — active low input to enable data input to UD0 – UD3.
11	$\overline{ME}$	Master Enable — active low input to enable the $\overline{IV}$ bus for data input, data output, or $\overline{IV}$ address selection/deselection; UD-bus operations are unaffected.
12	GND	Ground.
13	MCLK	Master Clock — active high input from Microcontroller used to strobe data into data latches from the $\overline{IV}$ bus and bits UD0 – UD3 of the UD bus; MCLK also synchronizes $\overline{IV}$ address selection.
14	SC	Select Command — active high input from Microcontroller to enable $\overline{IV}$ address input from the $\overline{IV}$ bus for device selection.
15	WC	Write Command — active high from Microcontroller to enable the writing of data into the data latches from the $\overline{IV}$ bus, provided $\overline{UIC}$ is not low.
16 – 23	$\overline{IV0}$ – $\overline{IV7}$	Interface Vector (Input/Output Bus) — 3-State bidirectional, data bus; $\overline{IV0}$ corresponds to UD0.
24	V_{CC}	Supply Voltage.

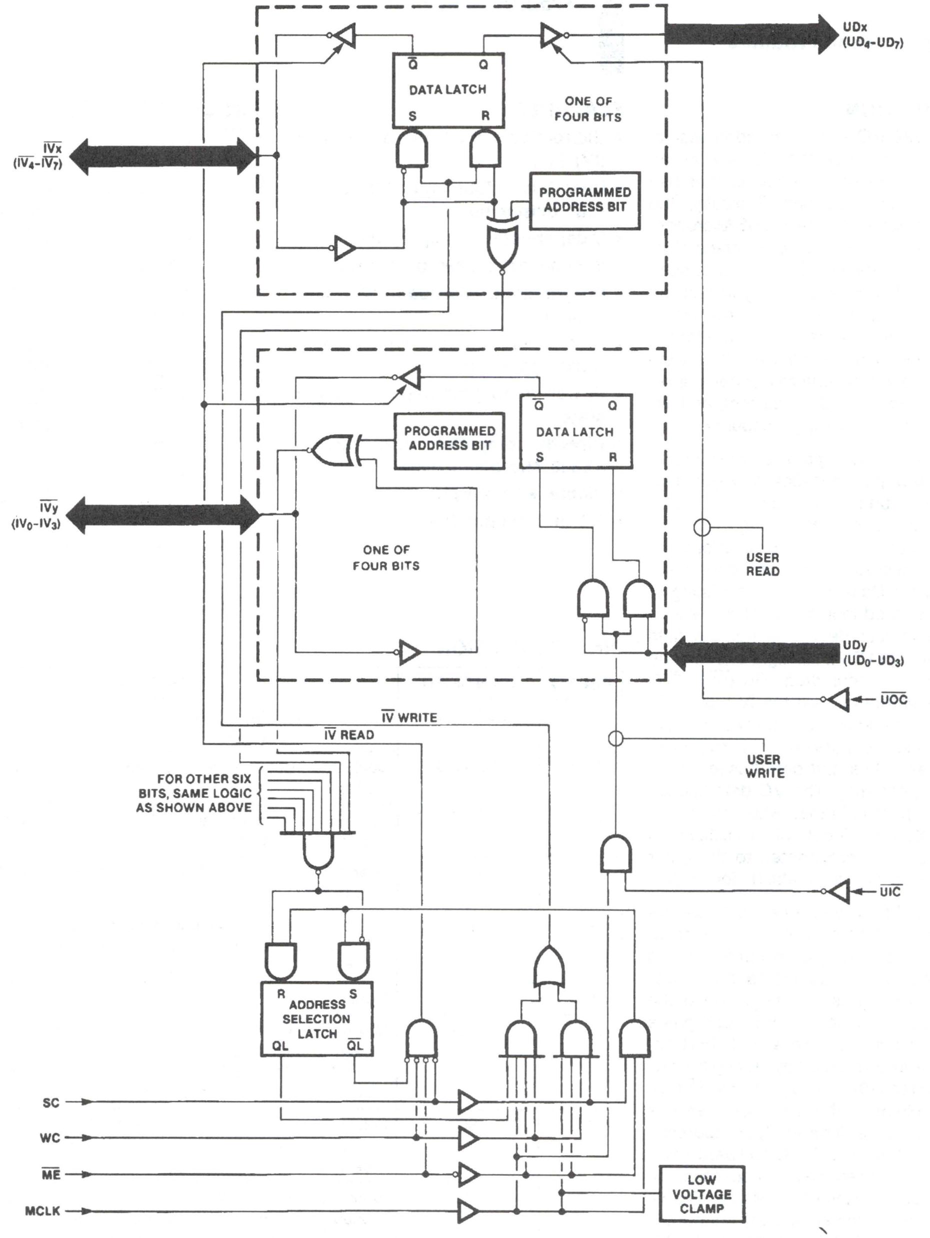

UDx
(UD4-UD7)
DATA LATCH
Q̄
Q
S
R
ONE OF
FOUR BITS
IVx
(IV4-IV7)
PROGRAMMED
ADDRESS BIT
PROGRAMMED
ADDRESS BIT
DATA LATCH
IVy
(IV0-IV3)
ONE OF
FOUR BITS
USER
READ
UDy
(UD0-UD3)
UOC
IV WRITE
IV READ
USER
WRITE
FOR OTHER SIX
BITS, SAME LOGIC
AS SHOWN ABOVE
UIC
ADDRESS
SELECTION
LATCH
QL
Q̄L
SC
WC
ME
MCLK
LOW
VOLTAGE
CLAMP
BD08631S

Figure 1. Logic Diagram for 8X382 I/O Port

4-Input/4-Output Addressable I/O Port 8X382

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data-bus interface is controlled by the $\overline{\mathrm{UIC}}$ and $\overline{\mathrm{UOC}}$ inputs. Data input to UD0 – UD3 is synchronous with MCLK, that is, with $\overline{\mathrm{UIC}}$ low, information is written into the data latches only when MCLK is high. The output drivers of UD4 – UD7 bus are enabled when $\overline{\mathrm{UOC}}$ is low.

$\overline{\mathrm{IV}}$ Bus Control

Input/output control of the $\overline{\mathrm{IV}}$ bus is shown in Table 2; this bus is controlled by SC, WC, $\overline{\mathrm{ME}}$, MCLK and the current state of an internal address selection latch. The address selection latch in the I/O port stores the result of the most recent $\overline{\mathrm{IV}}$ address selection. The latch is set when the internally preprogrammed address of the port matches the address on the $\overline{\mathrm{IV}}$ bus during an address-selection operation (SC = MCLK = High/ WC = Low). The latch is cleared when the two 8-bit address patterns are in disagreement. The $\overline{\mathrm{IV}}$ bus can transfer data only when the selection latch is set. The Microcontroller Left Bank ($\overline{\mathrm{LB}}$) and Right Bank ($\overline{\mathrm{RB}}$) outputs can control the $\overline{\mathrm{ME}}$ inputs for two banks of I/O devices, thus acting as ninth address bit.

Data is written into the data latches of a selected device from the $\overline{\mathrm{IV}}$ bus when WC = MCLK = High and $\overline{\mathrm{ME}}$ = Low. Output drivers on the $\overline{\mathrm{IV}}$ bus of the device with the address latch set are enabled with $\overline{\mathrm{ME}}$, WC, and SC low. With SC and WC both high, the bit pattern present on $\overline{\mathrm{IV0}}$ – $\overline{\mathrm{IV7}}$ is interpreted as both input data ($\overline{\mathrm{IV4}}$ – $\overline{\mathrm{IV7}}$ only) and $\overline{\mathrm{IV}}$ address. The data in $\overline{\mathrm{IV4}}$ – $\overline{\mathrm{IV7}}$ is latched in whether or not the I/O port has been previously selected. If the preprogrammed address of the I/O port matches the bit pattern on $\overline{\mathrm{IV0}}$ – $\overline{\mathrm{IV7}}$ when SC and WC are both high, the selection latch is set; otherwise it is reset. (Note. The Microcontroller never drives both SC and WC high at the same time.)

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (NOTE: A logic "1" in Microcontroller software corresponds to a high-level on the UD bus even though the $\overline{\mathrm{IV}}$ bus is inverted). The 8X382 wakes up in the unselected state with all data bits latched at the "logic 1" level (UD bus outputs high if enabled).

Table 1. Input/Output Control of UD Bus

$\overline{\mathrm{UIC}}$	$\overline{\mathrm{UOC}}$	MCLK	FUNCTION OF UD BUS	
			UD0 – UD3	UD4 – UD7
H	L	X	Inactive	Output Data
L	X	H	Input Data	Inactive
L	X	L	Inactive	Inactive
H	H	X	Inactive	Inactive

NOTE:
X = don't care

Table 2. Input/Output Control of IV Bus

$\overline{\mathrm{ME}}$	SC	WC	MCLK	SEL LATCH	FUNCTION OF $\overline{\mathrm{IV}}$ BUS
L	L	L	X	Set	Output Data
L	L	H	H	Set	Input Data ($\overline{\mathrm{IV4}}$ – $\overline{\mathrm{IV7}}$ only)
L	H	L	H	X	Input Address*
L	H	H	H	X	Input Data (IV4 – IV7 only) and address*
L	X	H	L	X	Inactive
L	H	X	L	X	Inactive
L	L	X	X	Not set	Inactive
H	X	X	X	X	Inactive

NOTES:
X = don't care
* Selection latch is updated

Table 3. Programming Specifications

PARAMETERS	LIMITS			UNITS
	Min	Typ	Max	
V_{CCP} — Programming supply voltage: Address	8.75	9	9.25	V
Protect		0		V
Maximum time $V_{CCP} > 5.25V$			1	Sec
Programming voltage: Address	8.75	9	9.25	V
Protect	8.75		9.25	V
Programming current: Address			5	mA
Protect			50	mA
t_r — Programming pulse/rise time: Address	10		100	µS
Protect	10		100	µS
t_ω — Programming pulse width	0.5		1	mS

ADDRESS PROGRAMMING AND ADDRESS PROTECT

Programming Procedures

The 8X382 can be programmed to respond to any address within a range of 0_{10} through 255_{10}. In an unprogrammed state, low-level ($\leqslant 0.8V$) inputs on all $\overline{\mathrm{IV}}$ bus lines (address 255_{10}) will select the device. To program a given address bit to match a high-level ($\geqslant 2V$) input on the corresponding $\overline{\mathrm{IV}}$ pin a (a logical "0" to the Microcontroller), the counterpart UD-bus pin must be pulsed according to Table 3 and the following procedures:

Step 1: Set all control inputs to the inactive state — $\overline{\mathrm{UIC}} = \overline{\mathrm{UOC}} = \overline{\mathrm{ME}} = V_{CC}$ and SC = WC = MCLK = GND; leave the UD and $\overline{\mathrm{IV}}$ bus pins open.

Step 2: Increase V_{CC} to V_{CCP}.

4-Input/4-Output Addressable I/O Port 8X382

Step 3: After V_{CC} has stabilized, apply a single programming pulse (Figure 2) to the user-bus bit that corresponds to the desired high-level $\overline{IV}$ address bit. The I/O port is programmed from the user bus (UD0 – UD7) for addressing from the Microcontroller bus (IV0 – IV7).

Step 4: Return V_{CC} to 0-volts. (Note: if the programming of all address bits is completed in less than 1-second, V_{CC} can remain at 9V for the required interval of time.)

Step 5: Steps 1 through 3 are applicable to the programming of each address bit that requires a high-level $\overline{IV}$ match.

Step 6: To verify that the address is properly programmed, return V_{CC} to +5V, set $\overline{IV0}$ – $\overline{IV7}$ to the desired (inverted) binary address pattern, set $\overline{ME}$ = WC = Low and SC = MCLK = High. If there are no programming errors, subsequent data written from $\overline{IV4}$ – $\overline{IV7}$ (WC = High) will appear inverted on UD4 – UD7.

Address Protect

After programming the I/O Port, steps should be taken to isolate the address circuits and make these circuits permanently immune to further change.

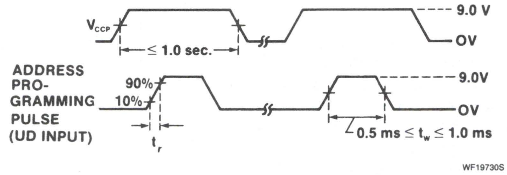


Figure 2. Address Programming Pulse

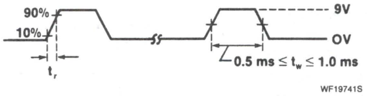


Figure 3. Protect Programming Pulse

Step 1: Set V_{CC} and all control inputs to 0-volts (V_{CC} = $\overline{UIC}$ = $\overline{UOC}$ = $\overline{ME}$ = SC = WC = MCLK = GND = 0.0V); $\overline{IV0}$ – $\overline{IV7}$ = open circuit.

Step 2: Taking one pin at a time, apply a protect programming pulse (Figure 3) to each user-bus bit (UD0 – UD7) — refer to Table 3 for min/max specifications pertaining to voltage and current.

Step 3: Verify that the address circuits for each bit is isolated by applying 9-volts, in turn, to each user-bus pin (UD0 – UD7) and measuring less than 200 microamperes of input current. (Note. Setup conditions are the same as those in Step 1.)

4-Input/4-Output Addressable I/O Port

8X382

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage[3]	+7	V_{DC}
V_{IN}	Input voltage[3]	+5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $4.75V \leqslant V_{CC} \leqslant 5.25V$, $0°C \leqslant T_A \leqslant 70°C$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{CC}	Supply voltage		4.75	5	5.25	V
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				0.8	V
V_{CL}	Input clamp voltage	V_{CC} = Min; $I_I = -10mA$			−1.5	V
I_{IH}	High-level input current[1]	V_{CC} = Max; $V_{IH} = 2.7V$		5	100	μA
I_{IL}	Low-level input current[1]	V_{CC} = Max; $V_{IL} = 0.5V$		−350	−550	μA
I_{OZH}	High-Z state output current — high level[4]	V_{CC} = Max; $V_{OH} = 2.5V$			100	μA
I_{OZL}	High-Z state output current — low level[4]	V_{CC} = Max; $V_{OL} = 0.5V$			−100	μA
V_{OL}	Low-level output voltage $\overline{IV}$ bus ($\overline{IV0}$ – $\overline{IV7}$)	V_{CC} = Min; $I_{OL} = 16mA$			0.55	V
	User bus (UD4 – UD7)	V_{CC} = Min; $I_{OL} = 24mA$			0.55	V
V_{OH}	High-level output voltage	V_{CC} = Min; $I_{OH} = -3.2mA$	2.4			V
I_{OS}	Short circuit output current[2] $\overline{IV}$ bus ($\overline{IV0}$ – $\overline{IV7}$)	V_{CC} = Max	−20			mA
	UD bus (UD4 – UD7)	V_{CC} = Max	−10			mA
I_{CC}	Supply current	V_{CC} = Max; $\overline{ME} = \overline{UOC} = V_{CC}$		90	150	mA

NOTES:

1. The input current includes the 3-State leakage current of the output driver on the data lines.
2. Only one output may be shorted at a time.
3. These limits do not apply during address programming.
4. Applies only to pins UD4 – UD7.

AC ELECTRICAL CHARACTERISTICS $4.5V \leqslant V_{CC} \leqslant 5.25V$, $0°C \leqslant 70°C$

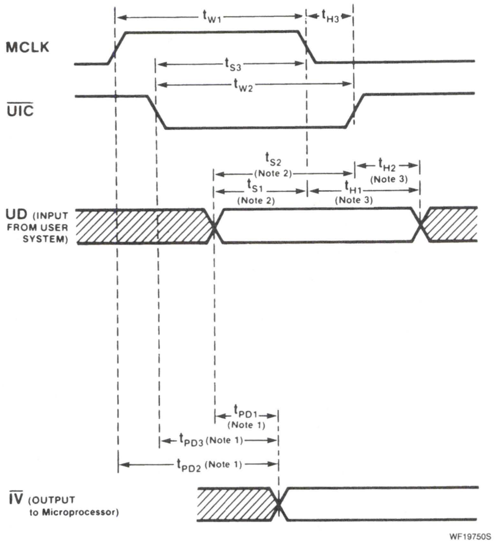
SYMBOL	PARAMETER	REFERENCES[1]		TEST CONDITIONS	LIMITS			UNIT
		From	To		Min	Typ	Max	
Pulse Widths								
t_{W1}	Clock High	↑ MCLK	↓ MCLK		35			ns
t_{W2}	User Input Control	↓ $\overline{UIC}$	↑ $\overline{UIC}$	MCLK = High	35			ns
Propagation Delays								
t_{PD1}	UD Propagation delay	UD_{0-3}	$\overline{IV}_{0-3}$	MCLK = High SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			30	ns
t_{PD2}	UD Clock delay	↑ MCLK	$\overline{IV}_{0-3}$	UD_{0-3} = Stable; MCLK = High; SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			50	ns
t_{PD3}	UD Input delay	↓ $\overline{UIC}$	$\overline{IV}_{0-3}$	UD_{0-3} = Stable; MCLK = High; SC = WC = $\overline{ME}$ = Low			50	ns
t_{PD4}	$\overline{IV}$ Data propagation delay	$\overline{IV}_{4-7}$	UD_{4-7}	MCLK = WC = High; $\overline{ME}$ = $\overline{UOC}$ = SC = Low			45	ns
t_{PD5}	$\overline{IV}$ Data clock delay	↑ MCLK	UD_{4-7}	$\overline{IV}_{4-7}$ = Stable; WC = High; $\overline{ME}$ = $\overline{UOC}$ = SC = Low			55	ns
Output Enable Timing								
t_{OE1}	UD Output enable	↓ $\overline{UOC}$	UD_{4-7}				30	ns
t_{OE3}	$\overline{IV}$ Data master enable	↓ $\overline{ME}$	$\overline{IV}$	WC = SC = Low			22	ns
t_{OE5}	$\overline{IV}$ Data write recovery	↓ WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			25	ns
t_{OE6}	$\overline{IV}$ Data select recovery	↓ SC	$\overline{IV}$	WC = $\overline{ME}$ = Low			25	ns
Output Disable Timing								
t_{OD1}	UD Output disable	↑ $\overline{UOC}$	UD_{4-7}				25	ns
t_{OD3}[2]	$\overline{IV}$ Data master disable	↑ $\overline{ME}$	$\overline{IV}$	WC = SC = Low			25	ns
t_{OD5}[2]	$\overline{IV}$ Data write override	↑ WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			20	ns
t_{OD6}[2]	$\overline{IV}$ Data select override	↑ SC	$\overline{IV}$	WC = $\overline{ME}$ = Low			20	ns

AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	REFERENCES[1]		TEST CONDITIONS	LIMITS			UNIT
		From	To		Min	Typ	Max	
Setup Times								
t_{S1}	UD Clock setup time	UD_{0-3}	↓ MCLK	$\overline{\text{UIC}}$ = Low	15			ns
t_{S2}	UD Control setup time	UD_{0-3}	↑ $\overline{\text{UIC}}$	MCLK = High	15			ns
t_{S3}	User input control setup time	↓ $\overline{\text{UIC}}$	↓ MCLK		25			ns
t_{S4}	$\overline{\text{IV}}$ Data setup time	$\overline{\text{IV}}$	↓ MCLK	WC = High or SC = High; $\overline{\text{ME}}$ = Low;	35			ns
t_{S5}[3]	$\overline{\text{IV}}$ Master enable setup time	↓ $\overline{\text{ME}}$	↓ MCLK	WC = High or SC = High	30			ns
t_{S6}	$\overline{\text{IV}}$ Write control setup time	↑ WC	↓ MCLK	SC = $\overline{\text{ME}}$ = Low;	30			ns
t_{S7}	$\overline{\text{IV}}$ Select control setup time	↑ SC	↓ MCLK	WC = $\overline{\text{ME}}$ = Low	30			ns
Hold Times								
t_{H1}	UD Clock hold time	↓ MCLK	UD	$\overline{\text{UIC}}$ = Low	15			ns
t_{H2}	UD Control hold time	↑ $\overline{\text{UIC}}$	UD	MCLK = High	15			ns
t_{H3}	User input control hold time	↓ MCLK	↑ $\overline{\text{UIC}}$		0			ns
t_{H4}	Data hold time	↓ MCLK	$\overline{\text{IV}}$	WC = High or SC = High; $\overline{\text{ME}}$ = Low	5			ns
t_{H5}[3]	Master enable hold time	↓ MCLK	↑ $\overline{\text{ME}}$	WC = High or SC = High;	0			ns
t_{H6}	$\overline{\text{IV}}$ Write control hold time	↑ MCLK	↓ WC	SC = $\overline{\text{ME}}$ = Low	0			ns
t_{H7}	$\overline{\text{IV}}$ Select control hold time	↑ MCLK	↓ SC	WC = $\overline{\text{ME}}$=Low	0			ns

NOTES:

1. All measurements to the $\overline{\text{IV}}$ bus assumes the address selection latch is set.
2. These parameters are measured with a capacitive loading of 50pF and represent the output driver turn-off time.
3. If $\overline{\text{ME}}$ is to be High (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing into or selection of the I/O port.


MCLK
$\overline{UIC}$
UD (INPUT FROM USER SYSTEM)
$\overline{IV}$ (OUTPUT to Microprocessor)
t_{W1}
t_{H3}
t_{S3}
t_{W2}
t_{S2} (Note 2)
t_{H2} (Note 3)
t_{S1} (Note 2)
t_{H1} (Note 3)
t_{PD1} (Note 1)
t_{PD3} (Note 1)
t_{PD2} (Note 1)
WF19750S

STATIC CONDITIONS:
$\overline{SC}$ = WC – $\overline{ME}$ – Low;
$\overline{UOC}$ = High

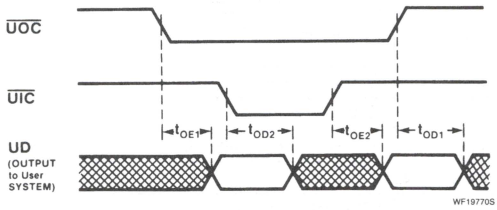
NOTES:
1. The actual time for stable data on the $\overline{IV}$ bus is the latest propagation from t_{PD1}, t_{PD2}, and t_{PD3}.
2. The UD input must satisfy the setup-time requirements for both t_{S1} and t_{S2}
3. Minimum hold-time required for the UD input is the earlier of the times specified by t_{H1} and t_{H2}.

a. User Data Input Timing

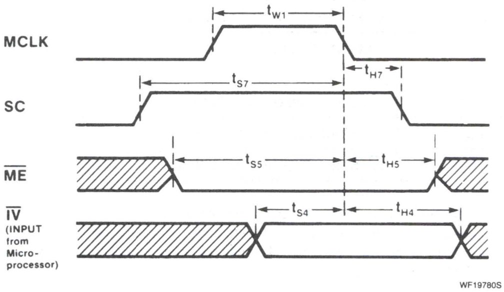
Legend:

▩ = 3-STATE

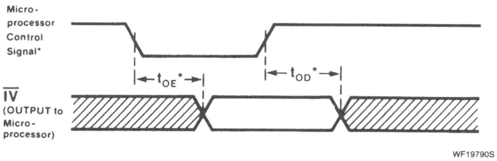
▨ = CHANGING DATA


$\overline{UOC}$
$\overline{UIC}$
UD (OUTPUT to User SYSTEM)
t_{OE1}
t_{OD2}
t_{OE2}
t_{OD1}
WF19770S

b. User Data Output Enable Timing


MCLK
SC
$\overline{ME}$
$\overline{IV}$ (INPUT from Micro-processor)
t_{W1}
t_{H7}
t_{S7}
t_{S5}
t_{H5}
t_{S4}
t_{H4}
WF19780S

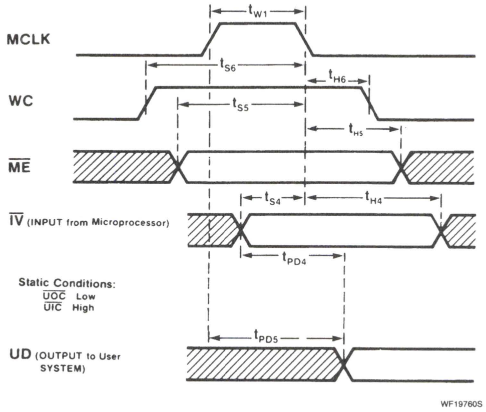
d. Microcontroller Select Cycle Timing


Micro-processor Control Signal*
$\overline{IV}$ (OUTPUT to Micro-processor)
t_{OE}*
t_{OD}*
WF19790S

***PARAMETER KEY**

MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS	STATIC CONDITIONS
$\overline{ME}$	t_{OE3} t_{OD3}	SC = WC = Low
WC	t_{OE5} t_{OD5}	SC = $\overline{ME}$ = Low
SC	t_{OE6} t_{OD6}	WC = $\overline{ME}$ = Low

e. Microcontroller Output Enable Timing

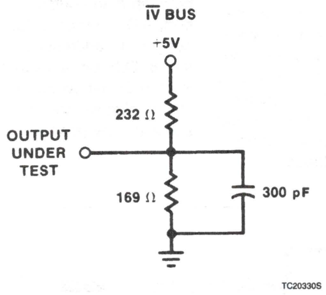

MCLK
WC
$\overline{ME}$
$\overline{IV}$ (INPUT from Microprocessor)
UD (OUTPUT to User SYSTEM)
t_{W1}
t_{S6}
t_{H6}
t_{S5}
t_{H5}
t_{S4}
t_{H4}
t_{PD4}
t_{PD5}
WF19760S

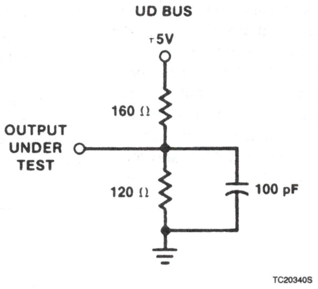
Static Conditions:
$\overline{UOC}$ Low
$\overline{UIC}$ High

c. Microcontroller Write Cycle Timing

Figure 4. Timing Diagrams

TEST LOADING CIRCUITS


$\overline{IV}$ BUS
+5V
232 Ω
OUTPUT UNDER TEST
169 Ω
300 pF
TC20330S


UD BUS
+5V
160 Ω
OUTPUT UNDER TEST
120 Ω
100 pF
TC20340S

PACKAGE DATA

Type: Plastic or Ceramic

Configuration: DIP

Width: 0.4 inches

Length: 1.2 inches

Pin Centers: 100 mils

Field-Programmable Addresses

Relevant ordering information for these parts is given below; unless otherwise indicated, referenced packaging data and operating specifications for parts with preassigned or field-programmable addresses are the same.

COMMERCIAL —
N8X382N (Plastic)
N8X3821 (Ceramic)

ORDERING INFORMATION

DESCRIPTION	ORDERING CODE	
	Preprogrammed Addresses	**Field Programmable Addresses**
24-Pin plastic DIP	N8X382N-ZZZ	N8X382N
24-Pin ceramic DIP	N8X382I-ZZZ	N8X382I

NOTE:
Where, ZZZ = any address from 000_{10} through 255_{10}. I/O Ports with preprogrammed addresses 0_{10} through 15_{10} are stock items; consult the nearest Signetics Sales and Service Office when ordering parts in the address range of 16_{10} through 255_{10}.

4-Input/4-Output Addressable I/O Port 8X382

APPLICATIONS

When compared to other Microcontroller ports in the 8X370 series, the 8X382 has some unique features that provide real design advantages in certain applications. Connection of the I/O port to the Microcontroller is simple and straightforward in that like pin names are tied together. The system designer must also decide on which bank of the Microcontroller to place the 8X382 and then connect the $\overline{\text{ME}}$ pin of the port to either the $\overline{\text{LB}}$ (Left Bank) or $\overline{\text{RB}}$ (Right Bank) of the Microcontroller.

The 8X382 is unique because it can be used for both dedicated input and output operations. In the system shown below, the user interface requires nine (9) dedicated inputs and eleven (11) dedicated outputs. Observe that by using an 8X382, the problem is solved by three devices, whereas four 8X372 ports are required for the same solution.

Another important use of the 8X382 is in implementing a handshake interface. Since both input and output bits reside in the same port, I/O operations can be performed without port re-addressing. Users may also find the 8X382 an advantage in the layout of Printed Circuit boards, since random control/status signals can be grouped within the same device position.

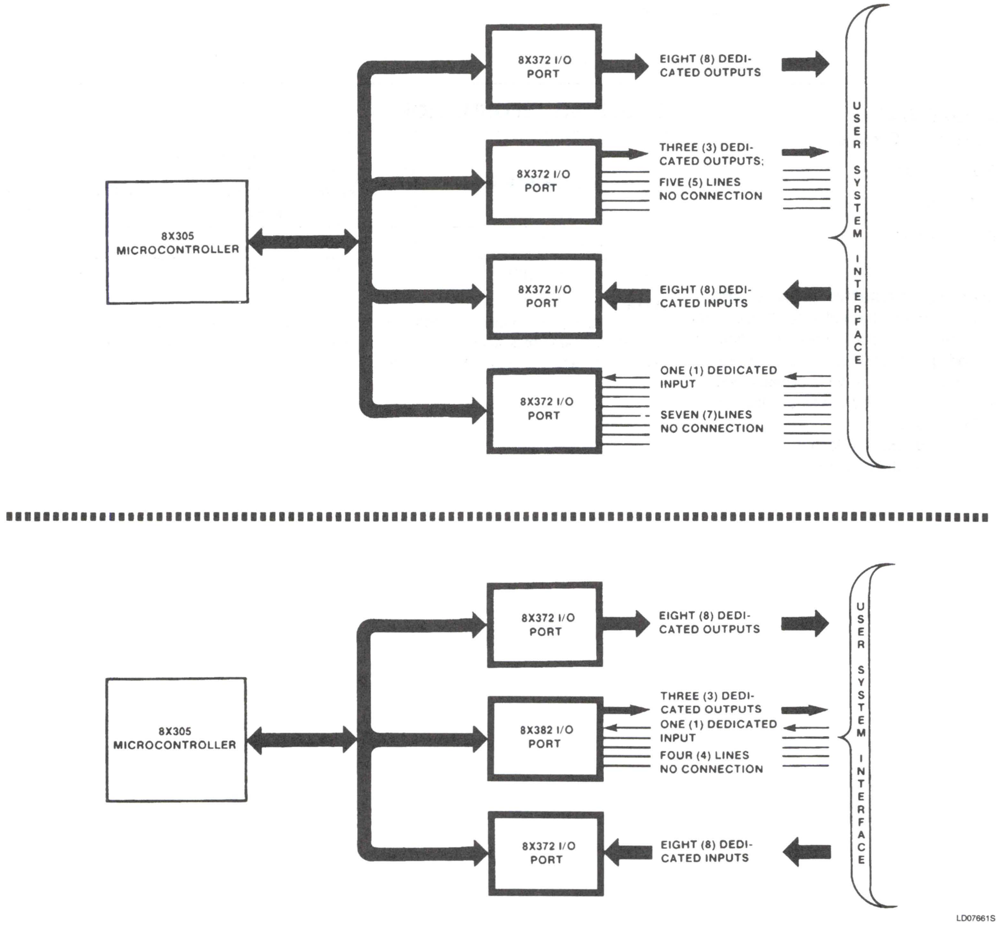

8X305 MICROCONTROLLER
8X372 I/O PORT
EIGHT (8) DEDICATED OUTPUTS
8X372 I/O PORT
THREE (3) DEDICATED OUTPUTS;
FIVE (5) LINES NO CONNECTION
8X372 I/O PORT
EIGHT (8) DEDICATED INPUTS
8X372 I/O PORT
ONE (1) DEDICATED INPUT
SEVEN (7)LINES NO CONNECTION
USER SYSTEM INTERFACE
8X305 MICROCONTROLLER
8X372 I/O PORT
EIGHT (8) DEDICATED OUTPUTS
8X382 I/O PORT
THREE (3) DEDICATED OUTPUTS
ONE (1) DEDICATED INPUT
FOUR (4) LINES NO CONNECTION
8X372 I/O PORT
EIGHT (8) DEDICATED INPUTS
USER SYSTEM INTERFACE
LD07661S

Figure 5. System Diagram for 8X382 I/O Port

Signetics

8X401
Microcontroller

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics 8X401 Microcontroller is a very high-speed bipolar microprocessor implemented with internal ECL technology. The 8X401 Microcontroller combines speed, flexibility, and a bit-oriented instruction set to accommodate many sophisticated applications. It excels in systems that require high-speed bit or byte manipulations, such as high-speed controllers and data communications.

The 8X401 can fetch, execute, and generate the next instruction address for a 20-bit instruction in a minimum of 150ns. Within one instruction cycle, the 8X401 can be programmed to input, right-rotate and mask single or multiple bit subfields, perform an ALU operation, left-rotate, merge the subfield into the destination, and output.

To interface with program memory, the 8X401 uses a 13-bit address bus and a 20-bit instruction bus. An 8-bit bidirectional data/address bus, and an I/O control and timing bus is used to access external peripheral devices.

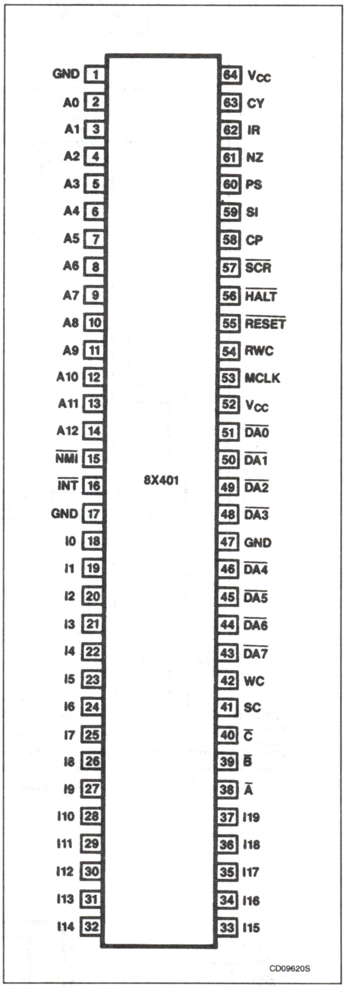
FEATURES

- **Fetches and executes all instructions in a minimum of 150ns**
- **Bit manipulation-oriented instruction set**
- **Separate buses for instruction, instruction address and I/O**
- **Sixteen 8-bit registers**
- **On-chip interrupt control**
- **TTL compatible I/O**
- **Single +5V supply**
- **0.9-inch 64-pin DIP, 68-pin plastic leaded chip carrier**
- **Two user-definable status flags**
- **Single TTL clock input**
- **Three independent I/O banks**
- **On-board control sequencer**
- **On-chip subroutine capabilities**
- **Fixed instruction set — 32 instructions**
- **Complete development support**

ORDERING INFORMATION

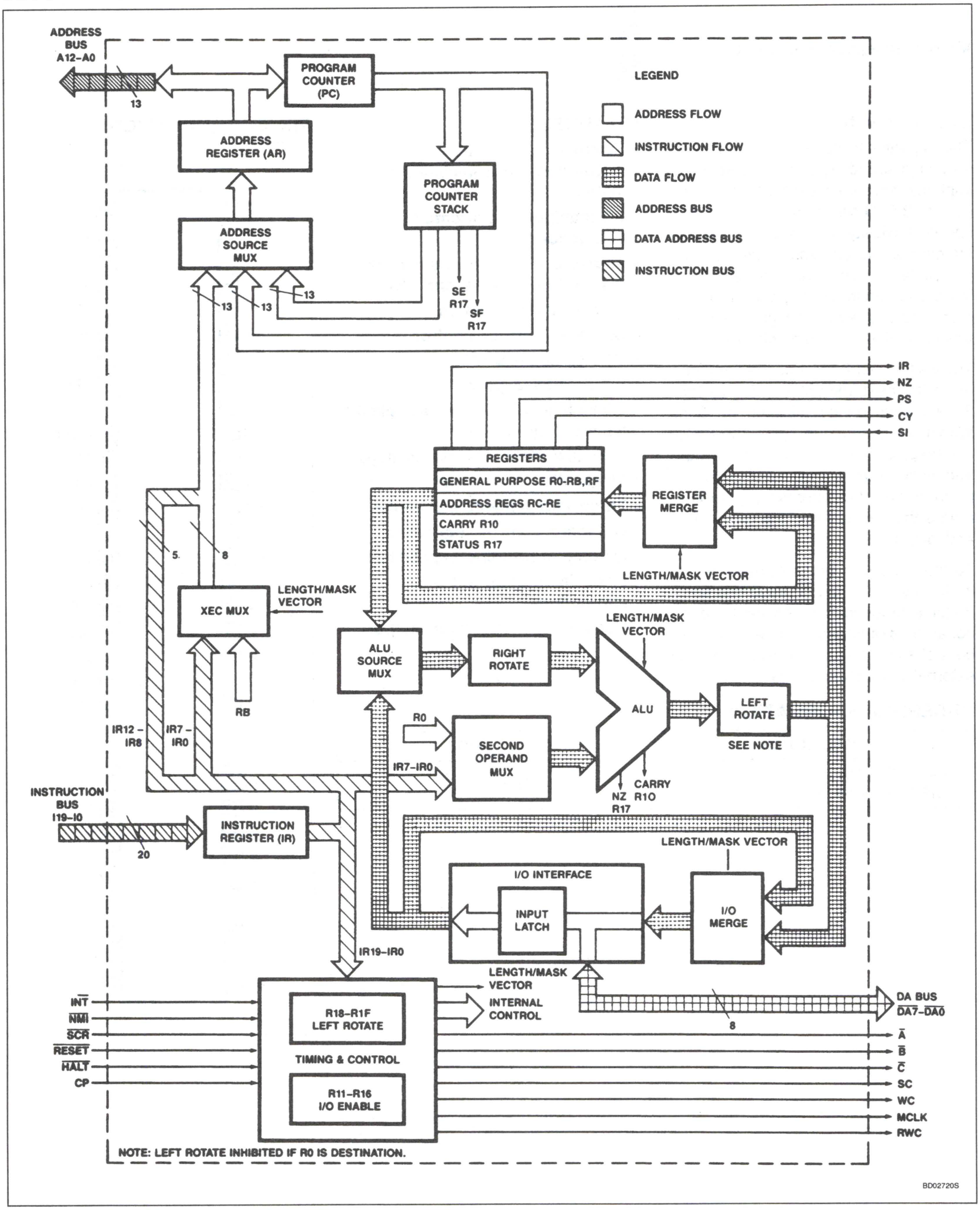
DESCRIPTION	ORDER CODE
64-pin CERAMIC DIP 900 Mil Wide	N8X401I

PIN CONFIGURATION



GND 1
A0 2
A1 3
A2 4
A3 5
A4 6
A5 7
A6 8
A7 9
A8 10
A9 11
A10 12
A11 13
A12 14
NMI 15
INT 16
GND 17
I0 18
I1 19
I2 20
I3 21
I4 22
I5 23
I6 24
I7 25
I8 26
I9 27
I10 28
I11 29
I12 30
I13 31
I14 32
64 Vcc
63 CY
62 IR
61 NZ
60 PS
59 SI
58 CP
57 SCR
56 HALT
55 RESET
54 RWC
53 MCLK
52 Vcc
51 DA0
50 DA1
49 DA2
48 DA3
47 GND
46 DA4
45 DA5
44 DA6
43 DA7
42 WC
41 SC
40 C
39 B
38 A
37 I19
36 I18
35 I17
34 I16
33 I15
8X401
CD09620S

BLOCK DIAGRAM OF THE 8X401


ADDRESS BUS A12–A0
13
PROGRAM COUNTER (PC)
ADDRESS REGISTER (AR)
PROGRAM COUNTER STACK
ADDRESS SOURCE MUX
13
13
13
SE R17
SF R17
LEGEND
ADDRESS FLOW
INSTRUCTION FLOW
DATA FLOW
ADDRESS BUS
DATA ADDRESS BUS
INSTRUCTION BUS
IR
NZ
PS
CY
SI
REGISTERS
GENERAL PURPOSE R0-RB,RF
ADDRESS REGS RC-RE
CARRY R10
STATUS R17
REGISTER MERGE
LENGTH/MASK VECTOR
5
8
XEC MUX
LENGTH/MASK VECTOR
RB
ALU SOURCE MUX
RIGHT ROTATE
LENGTH/MASK VECTOR
ALU
LEFT ROTATE
SEE NOTE
R0
SECOND OPERAND MUX
IR7–IR0
CARRY R10
NZ R17
IR12 – IR8
IR7 – IR0
INSTRUCTION BUS I19–I0
20
INSTRUCTION REGISTER (IR)
LENGTH/MASK VECTOR
I/O INTERFACE
INPUT LATCH
I/O MERGE
IR19–IR0
LENGTH/MASK VECTOR
INTERNAL CONTROL
DA BUS DA7–DA0
8
INT
NMI
SCR
RESET
HALT
CP
R18–R1F LEFT ROTATE
TIMING & CONTROL
R11–R16 I/O ENABLE
A
B
C
SC
WC
MCLK
RWC
BD02720S

NOTE: LEFT ROTATE INHIBITED IF R0 IS DESTINATION.

PIN DESCRIPTION

PIN NO.	IDENTIFIER	FUNCTION
1, 17, 47	GND	Ground.
2 – 14	A12 – A0	**Program Address Lines:** These active-high outputs permit direct addressing of up to 8192 locations of program storage; A0 is LSB.
15	$\overline{\text{NMI}}$	**Non-Maskable Interrupt:** The falling edge of this active-low input pin generates a non-maskable interrupt.
16	$\overline{\text{INT}}$	**Interrupt:** This active-low input pin is tested during the fourth quarter of each instruction cycle. If an interrupt is indicated and if interrupts are enabled, the address of the next instruction that was to be executed is stored onto the program counter stack before the interrupt is serviced.
18 – 37	I19 – I0	**Instruction Lines:** These active-high input lines receive 20-bit instructions from program storage; I0 is LSB.
38	$\overline{\text{A}}$	**Bank A:** When low, devices connected to bank A are accessed. (Note: Typically, the A signal is tied to the $\overline{\text{ME}}$ input pin of I/O peripherals.)
39	$\overline{\text{B}}$	**Bank B:** When low, devices connected to bank B are accessed. (Note: Typically, the B signal is tied to the $\overline{\text{ME}}$ input pin of I/O peripherals.)
40	$\overline{\text{C}}$	**Bank C:** When low, devices connected to bank C are accessed. (Note: Typically, the C signal is tied to the $\overline{\text{ME}}$ input pin of I/O peripherals.)
41	SC	**Select Control:** When high, an address is being output on pins $\overline{\text{DA7}}$ through $\overline{\text{DA0}}$.
42	WC	**Write Control:** When high, data is being output on pins $\overline{\text{DA7}}$ through $\overline{\text{DA0}}$.
43 – 46 48 – 51	$\overline{\text{DA7}}$ – $\overline{\text{DA0}}$	**Data Address Bus:** These active-low, bidirectional, three-state lines are used for I/O; $\overline{\text{DA0}}$ is LSB.
52, 64	V_{CC}	+5V power supply.
53	MCLK	**Master Clock:** This active-high output signal is used to strobe data into data peripherals for clocking I/O devices and/or synchronization of external logic. MCLK is active-high in the fourth quarter cycle.
54	RWC	**Read/Write Clock:** This active-high output signal is used for synchronization of external logic and is active-high during the third and fourth quarter cycles.
55	$\overline{\text{RESET}}$	**Reset:** The $\overline{\text{RESET}}$ input pin is used to initialize the 8X401.
56	$\overline{\text{HALT}}$	**Halt:** The $\overline{\text{HALT}}$ input is sampled during the first quarter cycle of each instruction cycle. When the $\overline{\text{HALT}}$ input is low, the instruction cycle is not executed.
57	$\overline{\text{SCR}}$	**Slow Clock Request:** This active-low control input is sampled during the first quarter cycle of each instruction. When $\overline{\text{SCR}}$ is asserted, it will cause the current instruction to be executed at half of the normal clock rate. This control input is necessary to accommodate I/O devices that cannot operate at the 8X401's full speed, without having to continuously run the 8X401 at half speed.
58	CP	**Clock Pulse:** Each 8X401 quarter cycle will correspond to one full cycle of the clock pulse.
59	SI	**Status Input:** The value of the SI pin during the fourth quarter cycle is transferred to SI bit in the status register.
60	PS	**Programmable Status:** The programmable status pin is controlled entirely by the user program.
61	NZ	**Non-Zero:** The NZ bit of the status register is reflected on this pin.
62	IR	**Interrupt Receivable:** The IR pin indicates whether an interrupt applied at any point in time will be serviced. Interrupts are receivable when the interrupt mask (status register, bit 0) is clear and the stack is not full (IM = 0 and SF = 0).
63	CY	**Carry:** Carry bit from R10 is output on this pin.

SMALL SYSTEM CONFIGURATION

The system hookup shown on the next page, although of the simplest form, provides a fundamental example of the 8X401 Microcontroller and compatible peripheral relationships. As shown, the 8X401 can directly address up to 8K locations of program storage.

Each of the three bank pins ($\overline{A}$, $\overline{B}$, or $\overline{C}$) are capable of uniquely addressing 256 input/output locations via the Data Address bus ($\overline{DA7}$ – $\overline{DA0}$).

The addressable locations for each bank can be used in a variety of ways. The hookup shown below is just one method of implementation.

When a particular bank signal is asserted, that bank is enabled and any one of 256 locations on that bank can be accessed for input/output operations.

PROGRAM STORAGE INTERFACE

As shown in the 8X401 small system hookup, program memory is connected to output address lines A12 through A0 (A0 = LSB) and input instruction lines I19 through I0 (I0 = LSB). An address output on A12 – A0 identifies one 20-bit instruction word in program memory. The program memory outputs an instruction word on I19 – I0 which defines the microcontroller operation which is to follow. One instruction word equals one completed operation. Any TTL-compatible memory can be used for program storage, provided the worst-case access time is compatible with the instruction cycle time used for the application. See timing section for appropriate calculations.

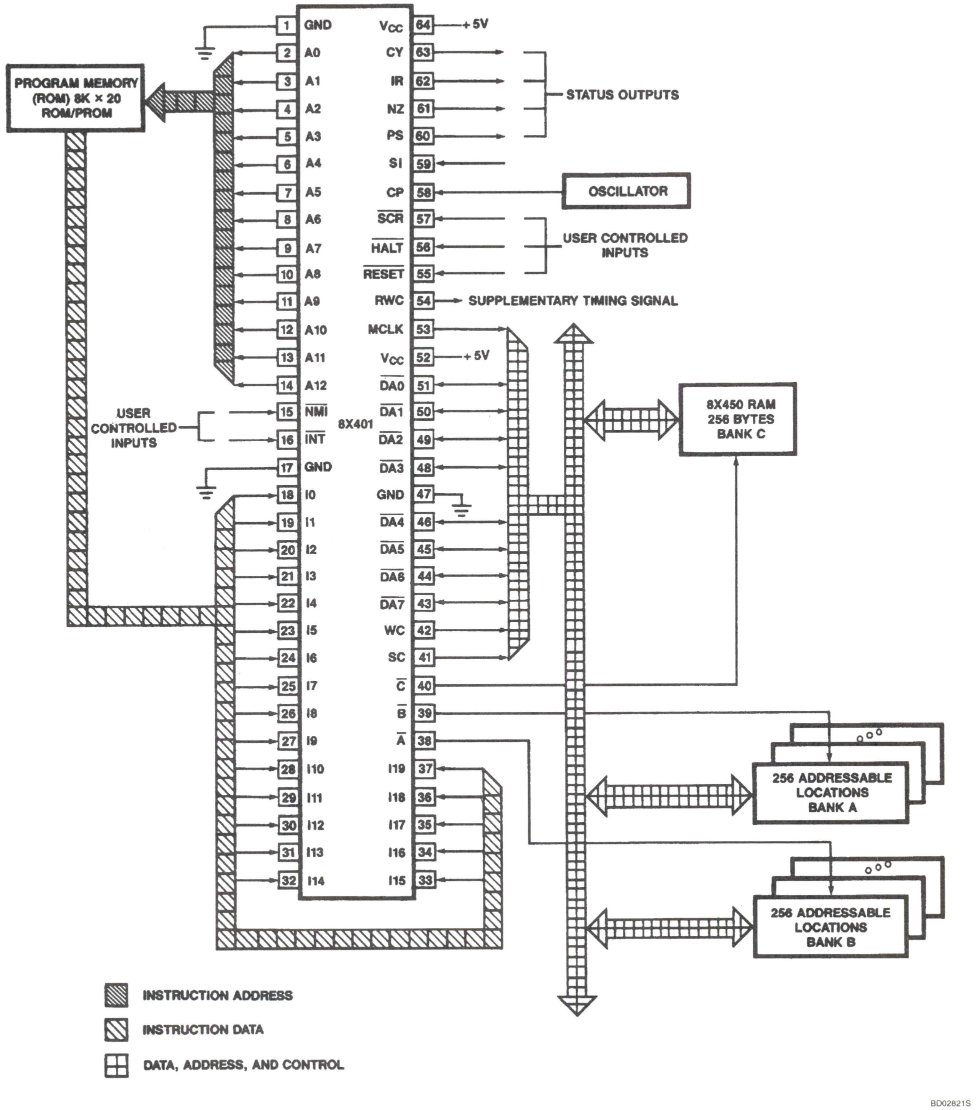
I/O INTERFACE AND CONTROL

The Data Address (DA) bus is an 8-bit bidirectional I/O bus which provides a communication link between the 8X401 and the three banks of the I/O devices. The $\overline{A}$ (A bank), $\overline{B}$ (B bank), and $\overline{C}$ (C bank) control signals identify which bank is enabled. When all three banks go high (inactive), neither bank is enabled and the DA bus is inactive (three-state). A functional analysis of the three bank signals is shown below:

A	B	C	FUNCTION
Low	Low	Low	This state is not generated by the 8X401.
Low	High	High	Enable A bank devices.
High	Low	High	Enable B bank devices.
High	High	Low	Enable C bank devices.
High	High	High	Disable all devices; DA bus is three-state

Both data and I/O address information are multiplexed on the DA bus. The SC (Select Command) and WC (Write Command) signals distinguish between data and I/O address information as shown in the table below. Although the table shows bank A only, the same conditions apply to banks B and C.

BANK A	SC	WC	FUNCTION
High	Low	Low	DA bus is three-state and not looking for input data.
Low	Low	Low	The DA bus is reading input data.
Low	Low	High	Data is being output.
Low	High	Low	Address is being output.
X	High	High	This condition is never generated.


1 GND
2 A0
3 A1
4 A2
5 A3
6 A4
7 A5
8 A6
9 A7
10 A8
11 A9
12 A10
13 A11
14 A12
15 NMI
16 INT
17 GND
18 I0
19 I1
20 I2
21 I3
22 I4
23 I5
24 I6
25 I7
26 I8
27 I9
28 I10
29 I11
30 I12
31 I13
32 I14
64 V_{CC} +5V
63 CY
62 IR
61 NZ
60 PS
59 SI
58 CP
57 SCR
56 HALT
55 RESET
54 RWC
53 MCLK
52 V_{CC} +5V
51 DA0
50 DA1
49 DA2
48 DA3
47 GND
46 DA4
45 DA5
44 DA6
43 DA7
42 WC
41 SC
40 C
39 B
38 A
37 I19
36 I18
35 I17
34 I16
33 I15
8X401
PROGRAM MEMORY (ROM) 8K × 20 ROM/PROM
USER CONTROLLED INPUTS
STATUS OUTPUTS
OSCILLATOR
USER CONTROLLED INPUTS
SUPPLEMENTARY TIMING SIGNAL
8X450 RAM 256 BYTES BANK C
256 ADDRESSABLE LOCATIONS BANK A
256 ADDRESSABLE LOCATIONS BANK B
▨ INSTRUCTION ADDRESS
▧ INSTRUCTION DATA
⊞ DATA, ADDRESS, AND CONTROL
BD02821S

8X401 Small System Hookup

DATA PROCESSING

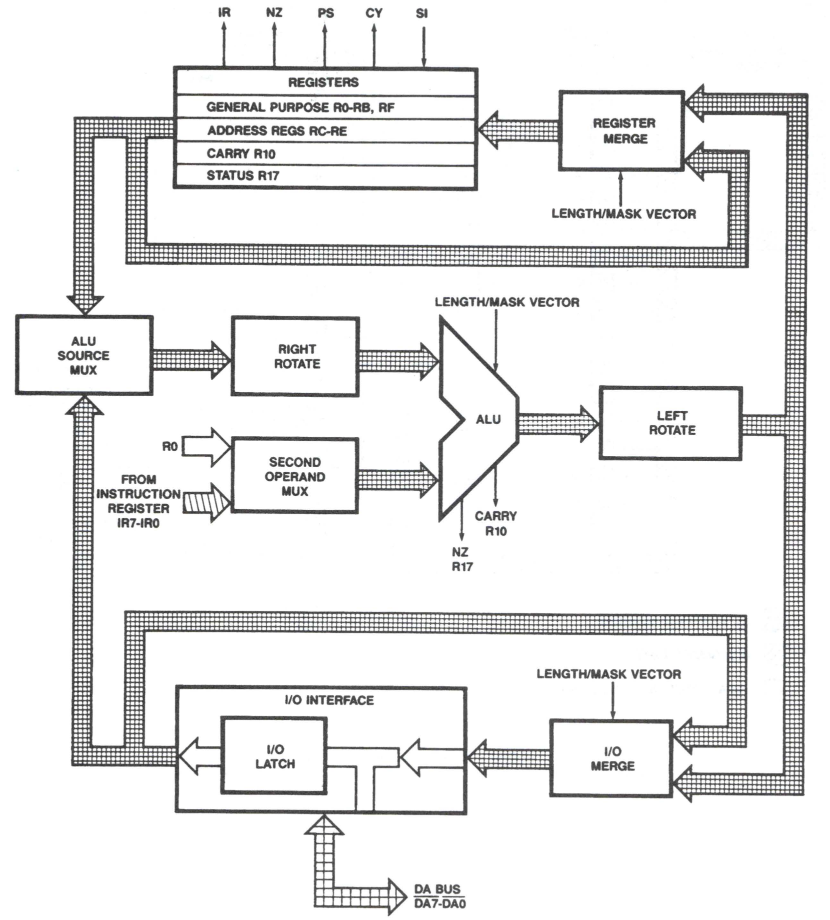
The data processing section of the 8X401 consists of a number of logical subsections. In order of processing, the data sees the right rotator, the ALU, the left rotator, and the merge circuits. Data sources and destinations can be various on-chip registers, the bidirectional DA bus, or immediate subfields. The data processing paths are shown below.

DATA REGISTERS

General-Purpose Storage

There are 13 source/destination general-purpose registers available on the 8X401. Three of these registers, specifically Registers 0, B, and F, have other special functions in addition to being general purpose. A summary of the registers is listed below:

- **R0 (Auxiliary Register)** — Register 0 is also used as the implied second operand for two operand instructions (ADD, ADD with CARRY, XOR, AND). The primary operand is specified in the source field of the instruction word and the AUX register is the implied second operand. Prior to performing arithmetic or logical operations (other than the IMMEDIATE operation), it is assumed that R0 contains the appropriate data. In order to reduce the possibility of erroneous results and to minimize the number of instructions required to transfer a right-justified second operand into the AUX, the left-rotate and merge functions are inhibited when specifying the AUX as a destination address. This allows subfields from any internal register or I/O bank to be transferred to the AUX with the subfield LSB right-justified and unspecified bits set to zero.
- **R1 through RA** — These 10 addresses specify general-purpose, on-chip storage registers.
- **RB** — Register B is also used as the implied source for the XEC instruction.
- **RF** — Register F is also used as the implied destination for the XOR IMMEDIATE and AND IMMEDIATE instruction classes.


IR
NZ
PS
CY
SI
REGISTERS
GENERAL PURPOSE R0-RB, RF
ADDRESS REGS RC-RE
CARRY R10
STATUS R17
REGISTER MERGE
LENGTH/MASK VECTOR
ALU SOURCE MUX
RIGHT ROTATE
LENGTH/MASK VECTOR
ALU
LEFT ROTATE
R0
FROM INSTRUCTION REGISTER IR7-IR0
SECOND OPERAND MUX
CARRY R10
NZ R17
LENGTH/MASK VECTOR
I/O INTERFACE
I/O LATCH
I/O MERGE
DA BUS DA7-DA0

BD02810S

Enabled I/O Addresses

These three registers (RC RD, and RE) always contain the address of the most recently enabled I/O device for each of the three I/O banks. When register C, D, or E is the specified destination address, the destination data is sent to both the on-chip register and the corresponding bank on the DA bus. The relationship between the register addresses and I/O banks is shown below:

REGISTER	BANK
C	A
D	B
E	C

When these registers are specified as a destination address, the L field must be set to 0 (full 8-bit operation). Also, note that registers C, D, and E may not be used with the XMIT 5 or ADD IMMEDIATE 5 instructions.

Carry

Register 10 contains the Carry bit. Bit position 0 (LSB) is the Carry bit, and positions one through seven are always zero. The Carry bit is updated each time an ADD, ADD IMMEDIATE, or ADD WITH CARRY instruction is performed. When specifying address 10 as a destination, only bit 0 (the Carry bit) will be written to. Data written to the Carry bit will be the LSB of the right-rotated data after any specified operation.

When the Carry register is the explicit destination of any ADD instruction, it will contain the carry resulting from the add operation rather than the LSB of the sum. Carry can also be affected via the Return and Set Carry or Return and Clear Carry instructions.

Status Register

This address specifies the current condition of the 8X401 system. The status register may be either a source or destination; however, certain bits in the status register are read-only. Four status outputs are available on 8X401 pins. They are NZ (Not Equal to Zero), PS (Programmable Status), IR (Interrupts Receivable), and Carry (R10, bit 0). The IR pin goes high when the interrupt mask is clear and the stack is not full. The IR output is updated during the 4th quarter cycle. The following descriptions define the bits within the status register.

Bit 0: (IM) — This bit represents the Interrupt Mask control. When IM is set, the interrupt is inhibited. This bit is set automatically by a response from a standard or non-maskable interrupt, or RESET. IM can also be set or cleared by a write to the status register.

Bit 1: (NZ) — This bit is set whenever the ALU output data is not equal to zero after any of the following instructions: MOVE, ADD, AND, XOR, ADD IMMEDIATE, AND IMMEDIATE, XOR IMMEDIATE, or ADD WITH CARRY. NZ can also be written to directly when specified in the destination field. This operation will negate and take priority over the normal setting by the ALU output. NZ is not affected after an XMIT instruction, or after a write to R17.

Bit 2: (PS) — This is the Programmable Status bit. The contents are reflected on the PS output pin. This status is controlled entirely by the user program.

Bits 3 and 4: (UF0, UF1) — These two bits represent user flags and have no assigned functions. They can be used as 1-bit internal flags and are entirely under control of the user program.

Bit 5: (SI) — This bit reflects the state of the status input pin. This read-only bit is updated during the 4th quarter cycle.

Bits 6 and 7: (SE, SF) — These read-only bits indicate Stack Empty and Stack Full, respectively. The bits are updated during the 3rd quarter cycle within the instruction that alters the stack status.

INSTRUCTION WORD (See Table 3)

Operations Code Field — The 4-bit opcode specifies one of 16 classes of instructions. Some instructions require two additional subopcode fields, X and XS. Variations and interpretations are displayed in Table 2.

Source (S) and Destination (D) Fields — The 5-bit "S" and "D" fields specify the source and destination, respectively, for the operation that is defined by the opcode. The "S" and/or "D" fields specify an internal 8X401 register or a variable length field from an I/O device. Hexadecimal values and source/destination field assignments for all internal registers are shown in Table 1.

When RC-RE (banks A, B, or C, respectively) are specified as the destination, the data is output onto the DA bus using the specified bank. The data is also stored in the specified register and may be later accessed as source data.

Rotate (R) and Length (L) Fields — The R field is used in conjunction with the L field to define the desired data within a register or I/O device. The source data is right-rotated prior to ALU operations, such that the bit specified by the R field is right-justified. The L field specifies the number of bits of data to be used for the operation. After the ALU operation, the data is left-rotated back to the original position prior to merging the data in the destination register.

When the L field specification is 0 (indicating a full 8-bit operation), the left-rotate is suppressed. This allows byte rotate operations to be performed. The left-rotate is also suppressed when the destination is register 0. This is the AUX register and is used as the implied second operand in certain instructions.

It should also be noted that subfields are defined at the ends of a register; for example, bit positions 1, 0, 7, and 6 constitute a contiguous 4-bit subfield.

Data Field — The data field holds data that can be processed directly from the instruction word.

LEFT-ROTATE OVERRIDE BLOCK

Register addresses 18-1F are destination only and are used to independently control left rotation of data prior to storage in the destination. Specifying 18-1F as a destination causes the data to be returned to the source address.

In order to move a processed subfield within the same register but in different bit positions (the LSB of the contiguous subfield can vary), it is necessary to independently specify the LSB for both the source and destination. The order of operation is as follows:

- Register or I/O source data is right-rotated as specified by the "R" field. Along with the "L" field, the subfield data is defined.
- Subfield data is processed via the ALU.
- Data is left-rotated 0 – 7 bits, depending on the corresponding register addresses 18-1F as specified in the destination field rather than using the "R" field.
- After left-rotation the specified subfield is merged into those bits of the original source data. The unspecified bits of the original source data remain unchanged.
- Result is stored in the register address specified by the source field.

Note that the left-rotate is always inhibited if the "L" field is zero. Also, addresses 18-1F may not be used in the destination field for the XMIT or ADD IMMEDIATE instructions. The destination addresses and corresponding left-rotate values are shown in Table 2.

DA BUS CONTROL BLOCK

Register addresses 11 – 16 are used by the 8X401 to access I/O devices for either a source or destination specified within the instruction. Register addresses 13, 15, and 16 specify banks A, B, and C, respectively, whereas addresses 11, 12, and 14 specify bank pairs AB, CA, and BC, respectively (Table 4). One bank of each pair is known as the preferred bank. The preferred banks for pairs AB, CA, and BC are banks A, C, and B, respectively. The first letter from each bank pair can serve as a mnemonic aid as to which bank is the preferred bank. Having a preferred bank is simply a method of determining which bank to read when an instruction would otherwise indicate that two banks should be read at once.

When used as a source, the appropriate I/O bank is enabled and data is read from the activated I/O device on that bank. The I/O device may have been activated by a previous address select instruction where registers C-E were the specified destination. If a bank pair (addresses 11, 12, or 14) is specified as a source, only data from the preferred bank of that pair will be read in.

When addresses 11 – 16 are specified as the destination address, the destination data is sent to the DA bus. The Write Control (WC) signal goes high, indicating data (as opposed to an address) is on the DA bus and is to be written to the activated I/O device on the selected bank(s).

When addresses 11 – 16 are specified as the destination and the "L" field is not zero, the following statements apply: If the source is a register and the destination is a single bank, the bank will be read (or the preferred bank of a bank pair will be read) to obtain the data required to perform the merge operation. The result is that processed data from the specified subfield of the source register is returned to that selected field of the destination bank and any bits outside of the specified subfield will be loaded with unprocessed data from the I/O device just read. If the destination is a bank pair, the data from the procedure just described is sent to both banks. Below is an example of the outcome of an ADD instruction with Length = 4, Rotate = 2, Source = R1, and Destination = R11 (bank pair AB).

Example 1:

OPCODE	L	S	R	D
ADD	4	R1	2	R11
				Bank Pair AB

	7	6	5	4	3	2	1	0
R1	a	b	c	d	e	f	g	h

Specified Subfield

Source register R1. Note specified subfield from L = 4 and R = 2.

R1 + R0	a'	b'	c'	d'	e'	f'	g'	h'

Add R1 with R0 (AUX). Result after left-rotate.

Bank A	i	j	k	l	m	n	o	p

Read preferred bank A of AB.

Bank A	i	j	c'	d'	e'	f'	o	p
Bank B	i	j	c'	d'	e'	f'	o	p

Result after merge. Data put out on both banks A and B.

If, however, the specified source is a bank or bank pair, any unspecified bits will contain unprocessed data from the source I/O device. Below is an example of the outcome with Source = R14 (bank pair BC) and Destination = R13 (bank A).

Example 2:

OPCODE	L	S	R	D
ADD	4	R14	2	R13
		Bank Pair BC		**Bank A**

	7	6	5	4	3	2	1	0
Bank B	a	b	c	d	e	f	g	h

Specified Subfield

Read Preferred bank B of BC. Note specified subfield from L = 4 and R = 2.

Bank B + R0	a'	b'	c'	d'	e'	f'	g'	h'

Add bank B data with R0 (AUX). Result after left-rotate.

Bank A	a	b	c'	d'	e'	f'	g	h

Result after merge. Unspecified bits contain original source data.

PROGRAM COUNTER STACK

The 8X401 stack is capable of saving up to four return addresses for subroutines and interrupts. Addresses are pushed onto the stack as a result of a call or a maskable or non-maskable interrupt. Addresses are popped from the stack as a result of an unconditional RETURN, a satisfied conditional RETURN, or a Pop Stack and Jump instruction. The status of the stack (whether empty, full, or neither) is available from the SE and SF flags in the status register and the Interrupts Receivable (IR) output pin.

INTERRUPTS

Interrupt ($\overline{\text{INT}}$)

The interrupt input is tested once each instruction cycle, during the fourth quarter cycle (see Figure 1). When the interrupt input is taken low and is enabled, the address of the next instruction is pushed onto the program counter stack.

Program flow is transferred to address 2 for the start of the service routine (Figure 2). This is accomplished by inserting a dummy instruction cycle after the interrupt is accepted.

The interrupt mask bit (R17, bit 0) is set automatically as part of the interrupt response.

The Interrupts Receivable (IR) pin indicates whether an interrupt applied at any point in time will be serviced. Interrupts are receivable when the interrupt mask is clear and the stack is not full (IM = 0 and SF = 0).

Non-Maskable Interrupt ($\overline{\text{NMI}}$)

The function of the non-maskable interrupt is similar to the standard interrupt, except that the interrupt receivable status has no effect on its operation and the address jumped to is 1 rather than 2 (Figure 2). Address 1 should contain an unconditional JUMP to the start of the NMI service routine. An NMI is triggered by a falling edge on the $\overline{\text{NMI}}$ input. The interrupt mask is set to prevent normal interrupts from interfering with the NMI service routine. Note that it may not always be possible to recover from an NMI, since the condition of the interrupt mask prior to the NMI is not known, and the NMI response may overflow the stack.

SLOW CLOCK REQUEST ($\overline{\text{SCR}}$)

This control input is sampled during the first quarter cycle of each instruction along with the instruction data. If the input is low, it will cause the current instruction to be executed at half of the normal clock rate. The purpose of this function is to facilitate accesses to I/O devices that cannot operate at the 8X401's full speed, without the need to run the 8X401 continuously at half speed.

$\overline{\text{HALT}}$

The $\overline{\text{HALT}}$ input is sampled during the first quarter cycle of each instruction. If the $\overline{\text{HALT}}$ input is low, the instruction cycle is not executed. The MCLK continues to operate normally (high every fourth quarter cycle), even though program execution has ceased. When the $\overline{\text{HALT}}$ input goes high, program execution will resume at the next falling edge of MCLK. The DA bus is also inactive during a HALT operation. Like MCLK, RWC continues to operate during a HALT operation.

$\overline{\text{RESET}}$ LOGIC

The $\overline{\text{RESET}}$ pin is used to initialize the 8X401. When $\overline{\text{RESET}}$ is low, the address outputs (A12 – A0) are high impedance, the stack pointer is set to the top of the stack (empty), MCLK is inhibited, RWC is low, and the interrupt mask (bit 0, register 17) is set.

When $\overline{\text{RESET}}$ is released, the address outputs all go low (program address 0). A dummy instruction cycle occurs to allow time to fetch the first instruction from program storage at address 0. Only MCLK, RWC, and the address bus are in operation during the dummy cycle. The first active instruction cycle will begin following the first MCLK after $\overline{\text{RESET}}$ is released. The instruction at address 0 should be an unconditional jump to the beginning of the main program (which may be proceeded by a power-up sequence to initialize the system (Figure 2).

If $\overline{\text{RESET}}$ is applied during program execution, its effect is immediate. That is, if MCLK is high, it may be prematurely terminated by $\overline{\text{RESET}}$.

Table 1. Hexadecimal Addresses and Source/Destination Specification

ADDRESS	DESIGNATION	S	D	ADDRESS	DESTINATION	S	D
00	R0 (AUX) — General Purpose[1]	X	X	10	R10 — Carry[6]	X	X
01	R1 — General Purpose	X	X	11	R11 — Bank Access Command (Bank Pair AB)	X	x
02	R2 — General Purpose	X	X	12	R12 — Bank Access Command (Bank Pair CA)	X	x
03	R3 — General Purpose	X	X	13	R13 — Bank Access Command (Bank A)	X	x
04	R4 — General Purpose	X	X	14	R14 — Bank Access Command (Bank Pair BC)	X	x
05	R5 — General Purpose	X	X	15	R15 — Bank Access Command (Bank B)	X	x
06	R6 — General Purpose	X	X	16	R16 — Bank Access Command (Bank C)	X	x
07	R7 — General Purpose	X	X	17	R17 — Status[4]	X	X
08	R8 — General Purpose	X	X	18	R18 — Suppress Left-Rotate[5]		X
09	R9 — General Purpose	X	X	19	R19 — Left-Rotate 1 Place[5]		X
0A	RA — General Purpose	X	X	1A	R1A — Left-Rotate 2 Places[5]		X
0B	RB — General Purpose[2]	X	X	1B	R1B — Left-Rotate 3 Places[5]		X
0C	RC — Address Reg (Bank A)	X	X	1C	R1C — Left-Rotate 4 Places[5]		X
0D	RD — Address Reg (Bank B)	X	X	1D	R1D — Left-Rotate 5 Places[5]		X
0E	RE — Address Reg (Bank C)	X	X	1E	R1E — Left-Rotate 6 Places[5]		X
0F	RF — General Purpose[3]	X	X	1F	R1F — Left-Rotate 7 Places[5]		X

NOTES:

1. Also used as implied second operand for two operand instructions.
2. Also used as implied source for XEC instructions.
3. Also used as implied destination for XOR IMMEDIATE and AND IMMEDIATE instructions.
4. Certain bits in the status register are read-only. (See Status Register within text.)
5. The result is returned to the register address specified by the source field.
6. Carry register, bit 0 is the carry bit. Bits 1 – 7 are always set to zero.

Microcontroller

8X401

Table 2. Various of Instruction Types

INSTRUCTION TYPE	VARIATION	OPCODE	X	XS	DESCRIPTION
MOVE	MOV	0000	–	–	MOVE
ADD	ADD	0001	–	–	ADD
	ADC	0101	–	–	ADD with CARRY
	AD8	1000	–	–	ADD IMMEDIATE 8
	AD5	1001	–	–	ADD IMMEDIATE 5
AND	AND	0010	–	–	AND
	AN8	1010	–	–	AND IMMEDIATE 8
	AN5	1011	–	–	AND IMMEDIATE 5
XOR	XOR	0011	–	–	Exclusive-OR
	XR8	1100	–	–	Exclusive-OR IMMEDIATE 8
	XR5	1101	–	–	Exclusive-OR IMMEDIATE 5
XEC	XEC	0100	–	–	EXECUTE
XMIT	XT8	0110	–	–	Transmit IMMEDIATE 8
	XT5	0111	–	–	Transmit IMMEDIATE 5
RETURN	RIF NS	1110	000	–	RETURN IF SI = 0
	RIF S	1110	001	–	RETURN IF SI = 1
	RIF NC	1110	010	–	RETURN IF CARRY = 0
	RIF C	1110	011	–	RETURN IF CARRY = 1
	RIF Z	1110	100	–	RETURN IF ALU = 0
	RIF NZ	1110	101	–	RETURN IF ALU ≠ 0
	PSJ	1110	110	–	POP STACK and JUMP
	RTN	1110	111	00	RETURN
	RCC	1110	111	10	RETURN and CLEAR CARRY
	RSC	1110	111	11	RETURN and SET CARRY
JUMP	JIF NS	1111	000	–	JUMP IF SI = 0
	JIF S	1111	001	–	JUMP IF SI = 1
	JIF NC	1111	010	–	JUMP IF CARRY = 0
	JIF C	1111	011	–	JUMP IF CARRY = 1
	JIF Z	1111	100	–	JUMP IF ALU = 0
	JIF NZ	1111	101	–	JUMP IF ALU ≠ 0
	JSR	1111	110	–	JUMP to SUBROUTINE
	JMP	1111	111	–	JUMP

Instruction Set Overview: The 8X401 instruction set is summarized in Table 2. Subsets of each instruction type are grouped together showing the variations of each instruction type. The hardware and software descriptions can be found in the data operations section.

Table 3. 8X401 Instruction Formats

1. Format for the MOVE, ADD, XOR, and ADD with Carry Instructions:

OPCODE	L	SOURCE	R	DESTINATION

2. Format for the XMIT 8 and ADD Immediate 8 Instruction:

OPCODE	L	DESTINATION	DATA8

3. Format for the XMIT 5 and ADD Immediate 5 Instruction:

OPCODE	L	DESTINATION	R	DATA5

4. Format for the AND Immediate 8, and XOR Immediate 8 Instruction:

OPCODE	L	SOURCE	DATA8

5. Format for the AND Immediate 5 and XOR Immediate 5 Instruction:

OPCODE	L	SOURCE	R	DATA5

6. Format for the JUMP, Subroutine Jump, Conditional Jump, and Conditional Return Instruction:

OPCODE	X	ADDRESS

7. Format for the Unconditional Return, Return and Set Carry, and Return and Clear Carry Instruction:

OPCODE	X	XS	(UNUSED

8. Format for the XEC Instruction:

OPCODE	L	ADDRESS

Table 4. I/O Acess Register

REGISTER	INPUT BANK	OUTPUT BANK
11	A	A & B
12	C	C & A
13	A	A
14	B	B & C
15	B	B
16	C	C

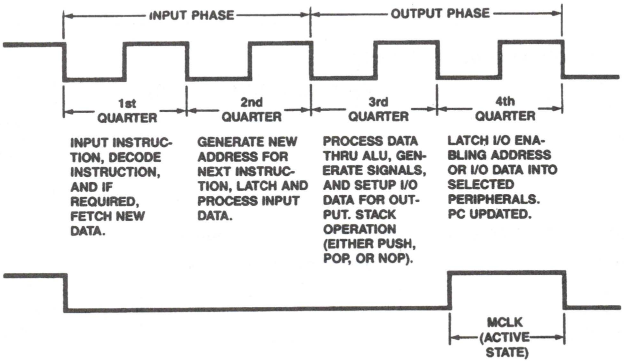

INPUT PHASE
OUTPUT PHASE
1st QUARTER
2nd QUARTER
3rd QUARTER
4th QUARTER
INPUT INSTRUCTION, DECODE INSTRUCTION, AND IF REQUIRED, FETCH NEW DATA.
GENERATE NEW ADDRESS FOR NEXT INSTRUCTION, LATCH AND PROCESS INPUT DATA.
PROCESS DATA THRU ALU, GENERATE SIGNALS, AND SETUP I/O DATA FOR OUTPUT. STACK OPERATION (EITHER PUSH, POP, OR NOP).
LATCH I/O ENABLING ADDRESS OR I/O DATA INTO SELECTED PERIPHERALS. PC UPDATED.
MCLK (ACTIVE STATE)
WF15240S

Figure 1. Instruction Cycle and MCLK with: Clock Input = 26.67MHz and Cycle Time = 150ns

DATA OPERATIONS OF THE 8X401 (See Tables 2 and 3)

MOVE, ADD, AND, XOR, ADD with CARRY

SOURCE	PRE-ALU	ALU	POST-ALU	DESTINATION
R0 – R17 as specified by "S" field of instruction.	Right rotate as specified by "R" field of instruction.	Perform appropriate ALU operation. (Note 1)	Left-rotate and mask as specified by the "R" and "L" fields of instruction.	R0 – R17 as specified by the "Destination" field of instruction. (Notes 2 & 3)

MOVE, ADD, AND, XOR, ADD with CARRY (Using left-rotate override R18 – R1F)

SOURCE	PRE-ALU	ALU	POST-ALU	DESTINATION
R0 – R17 as specified by "S" field of instruction.	Right-rotate as specified by "R" field of instruction.	Perform appropriate ALU operation. (Note 1)	Left-rotate and mask as specified by the "D" and "L" fields of instruction. (Note 4)	When R18 – R1F are specified in the "D" field, data is returned to the address specified in the "Source" field.

XMIT 8 (XT8), ADD IMMEDIATE 8 (AD8), AND IMMEDIATE 8 (AN8), XOR IMMEDIATE 8 (XR8)

SOURCE	PRE-ALU	ALU	POST-ALU	DESTINATION
One to eight bit constants from data field of instruction and register/address specified by either the source or destination field.	NOP.	Perform appropriate ALU operation. (Note 5)	Mask as specified by the "L" field.	**XT8, AD8:** Results are sent to register/address, starting from LSB, specified by the destination field of the instruction word. **AN8, XR8:** Results are sent to RF, starting from LSB.

XMIT 5 (XT5), ADD IMMEDIATE 5 (AD5), AND IMMEDIATE 5 (AN5), XOR IMMEDIATE 5 (XR5)

SOURCE	PRE-ALU	ALU	POST-ALU	DESTINATION
One to five bit constants from data field of instruction and register/address specified by either the source or destination.	Right-rotate as specified by "R" field, the data defined in either the source or destination field.	Perform appropriate ALU operation. (Note 6)	Left-rotate and mask as specified by the "R" and "L" fields.	**XT5, AD5:** Subfield is merged into register/address specified by destination field. LSB of data field is sent to bit position of destination defined by "R" field. **AN5, XR5:** Subfield is merged into register/address specified by source field. The "R" field specifies the LSB position of source and RF. Results are returned to RF.

ALL CONDITIONAL AND UNCONDITIONAL RETURNS

ADDRESS REGISTER (AR)	PROGRAM COUNTER (PC)	STACK
Conditional Returns: If condition is true. AR = address from top of stack, else load address from instruction word. **Unconditional Returns:** AR = address from top of stack.	PC = AR.	**Conditional Returns:** If condition is true. POP stack, else NOP. **Unconditional Returns:** POP stack.

ALL CONDITIONAL JUMPS, POP STACK AND JUMP (PSJ), and JUMP (JMP)

ADDRESS REGISTER	PROGRAM COUNTER	STACK
Conditional Jumps: If condition is true. AR = instruction word address, else AR = PC + 1. **PSJ and JMP:** AR = instruction word address.	PC = AR.	**Conditional Jumps:** NOP. **PSJ:** POP stack. **JMP:** NOP.

DATA OPERATIONS OF THE 8X401 (Continued)

CALL (JSR), INTERRUPT ($\overline{INT}$), NON-MASKABLE INTERRUPT ($\overline{NMI}$)

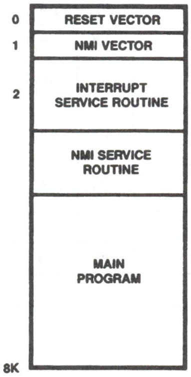
ADDRESS REGISTER	STACK	PROGRAM COUNTER
JSR: AR = instruction address. **$\overline{INT}$:** AR = address 2. **$\overline{NMI}$:** AR = address 1.	PUSH PC + 1. (Note 6)	PC = AR.

XEC

ADDRESS REGISTER	PROGRAM COUNTER	STACK
Right-most L bits of RB merged into corresponding bits in instruction address	PC not updated.	NOP.

NOTES:

1. **ALU Descriptions:**
   - MOVE: • No operation
   - ADD: • Source data ADDed to contents of auxiliary register (R0 – AUX). Carry bit set if carry is generated at MSB of selected data field. NZ status bit set if specified bits are not zero after ALU add.
   - AND: • Source data ANDed to contents of AUX register. NZ status bit updated accordingly.
   - XOR: • Source data Exclusive-ORed with contents of AUX register. NZ status bit set accordingly.
   - ADD with CARRY: • Sum is formed from source data. AUX register, and carry bit (register 10, bit 0). Carry and NZ status bits are set when appropriate.
2. Left-rotate is suppressed when destination is R0 (AUX).
3. When address registers RC, RD and RE are specified in the destination, source data will also go out on banks A, B, C, respectively. The L-field should be zero (a full 8-bit operation) to ensure duplication of the two outputs.
4. A left-rotate of 0 – 7 bits will correspond to R18 – R1F as specified in the "Destination" field of instruction word.
5. **ALU Descriptions:**
   - XMIT: • Input constants from the instruction word to specified destination. NZ flag is not updated when an XMIT is performed: however, NZ can be written to by an XMIT if R17 bit 1 is within the destination field.
   - ADD IMMEDIATE: • Instruction word data is ADDed to data specified by destination fied. The carry bit is set if a carry is generated at the MSB of the selected data field. NZ status bit is updated to reflect the value of "L" bits of data after the addition.
   - AND IMMEDIATE: • Instruction word data is ANDed to data specified by source field. Returning the destination data to RF allows the operation to be performed without destroying the original data field. This will facilitate testing of data for certain pre-defined values while still preserving the original data for other uses. NZ status bit updated accordingly. Unspecified bits in RF remain unchanged.
   - XOR IMMEDIATE: • Same as AND IMMEDIATE, except the logical operation performed is Exclusive-OR.
6. Note that the stack operation is show before the PC in the CALL and INTERRUPT formats. This is because the stack is actually in operation in cycle 3, and the PC is updated in cycle 4 (see Figure 1). In fact, for the Call (JSR) instruction and interrupt servicing, cycle order is important for the user to understand the current status of the PC. The other instructions are in reverse order for visual simplicity in keeping with block diagram flow, and cycle order is irrelevant.



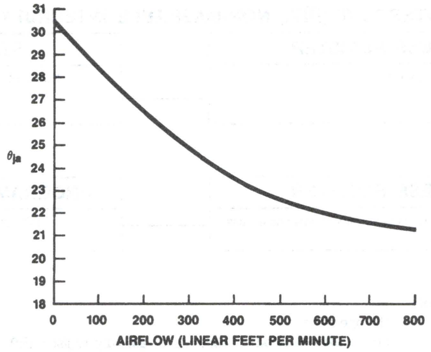
- THE RESET VECTOR, LOCATED AT ADDRESS 0, WILL BE AN UNCONDITIONAL JUMP TO THE BEGINNING OF THE MAIN PROGRAM.
- THE NON-MASKABLE INTERRUPT ($\overline{NMI}$) VECTOR, LOCATED AT ADDRESS 1, IS A JUMP TO THE NMI SERV-SERVICE ROUTINE.
- LOCATED AT ADDRESS 2 IS THE ROUTINE THAT SERVICES INTERRUPT ($\overline{INT}$) CALLS.

AF03650S

Figure 2. Typical 8X401 System Memory Map

Thermal Junction Temperature vs Airflow

The ceramic package used for the 8X401 has no heat sink and a θ_{ja} rating of 30.5°C/W in still air. Currently, to ensure operation at 150ns, the junction temperature of the devices must be kept below 115°C. The maximum power dissipation at that junction temperature will be 2.4W so that airflow will be required for full commercial range operation. The θ_{ja} versus Airflow curve is drawn here:



OP03880S

DC ELECTRICAL CHARACTERISTICS Commercial Part 4.75V ≤ V_{CC} ≤ 5.25V, 0°C ≤ T_A ≤ 70°C[1]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT	COMMENTS
			Min	Typ	Max		
V_{CC}	Supply voltage		4.75	5	5.25	V	
V_{IH}	High-level input voltage		2			V	
V_{IL}	Low-level input voltage				0.8	V	
V_{OH}	High-level output voltage	V_{CC} = Min, I_{OH} = −3mA	2.4			V	$\overline{\text{DA0}}$ through $\overline{\text{DA7}}$, MCLK, SC, WC, AB, BB, CB
		V_{CC} = Min, I_{OH} = −400μA	2.4			V	All others
V_{OL}	Low-level output voltage	V_{CC} = Min, I_{OL} = 16mA			0.5	V	$\overline{\text{DA0}}$ through $\overline{\text{DA7}}$, MCLK, RWC, SC, WC, $\overline{\text{A}}$, $\overline{\text{B}}$, $\overline{\text{C}}$
		V_{CC} = Min, I_{OL} = 8mA			0.5	V	A0 through A12, PS, NZ, CY, IR
V_{IC}	Input clamp voltage	V_{CC} = Min, I_{IN} = −10mA			−1.5	V	
I_{IH}	High-level input current	V_{CC} = Max, V_I = 2.7			20	μA	
I_{IL}	Low-level input current	V_{CC} = Max, V_I = 0.4V			−400	μA	
I_{OZH}	Off-state output current, high-level voltage applied	V_{CC} = Max, V_O = 2.7V			50	μA	$\overline{\text{DA0}}$ through $\overline{\text{DA7}}$
I_{OZL}	Off-state output current low-level voltage applied	V_{CC} = Max, V_O = 0.4V			−400	μA	$\overline{\text{DA0}}$ through $\overline{\text{DA7}}$
I_{OS}	Short circuit output current[2]	V_{CC} = Max, V_O = 0V	−30		−140	mA	
I_{CC}	Supply Current	V_{CC} = Max			500	mA	T_A = 0°C; Cold start[3]
					430	mA	T_J = 115°C

NOTES:

1. 64 Pin CDIP, Airflow required for full Commercial operation. (A plastic 64 Pin DIP with internal heat sink is currently under development which will not have this requirement.) See above for thermal characteristics.
2. Not more than one output should be tested at a time.
3. Guaranteed by Correlation to I_{CC} measured at 25°C.

AC ELECTRICAL CHARACTERISTICS ($V_{CC} = 5V +5\%$, $0°C \leqslant T_A \leqslant 70°C$)[1]

SYMBOL	PARAMETER	150ns CYCLE			> 150ns CYCLE			UNIT	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
t_{PC}	Processor cycle time	150						ns	
t_{CP}	Clock pulse period	37.5						ns	
t_{CH}	Clock pulse high time	15						ns	See Note 2
t_{CL}	Clock pulse low time	15						ns	
t_{MCL}	CP1 to MCLK low			30				ns	See Note 2
t_{MCH}	CP4 to MCLK high			40				ns	
t_W	MCLK pulse width	25	31	38	T4Q – 12		T4Q	ns	
t_{RWCL}	CP1 to RWC low		34	40				ns	
t_{RWCH}	CP3 to RWC high			40				ns	
t_{RWCW}	RWC pulse width	65	75	80	T3Q + T4Q – 10		T3Q + T4Q + 5	ns	
t_{AS}	CP2 to address stable			52				ns	
t_{MAS}	MCLK to address stable			62			T1Q + 24	ns	
t_{IS}	Instruction setup to CP1	0						ns	See Note 3
t_{MIS}	Instruction setup to MCLK	25						ns	
t_{IH}	Instruction hold from CP2	20						ns	
t_{MIH}	Instruction hold from MLCK	25			T1Q – 12			ns	
t_{SCH}	CP3 to SC rising edge			45				ns	
t_{MSCH}	MCLK to SC rising edge			95			T1Q + T2Q + 20	ns	
t_{WCH}	CP3 to WC rising edge			55				ns	
t_{MWCH}	MCLK to WC rising edge			105			T1Q + T2Q + 20	ns	
t_{WL}	CP1 to SC/WC falling edge			35				ns	See Note 4
t_{MWL}	MCLK to SC/WC falling edge	0						ns	
t_{IBSL}	CP1 to input phase bank signal falling edge			60				ns	
t_{MIBSL}	MCLK to input phase bank signal falling edge			33				ns	
t_{IBSH}	CP3 to input phase bank signal rising edge			45				ns	
t_{MIBSH}	MCLK to input phase bank signal rising edge			95			T1Q + T2Q + 20	ns	
t_{OBSL}	CP3 to output phase bank signal falling edge			53				ns	
t_{MOBSL}	MCLK to output phase bank signal falling edge			105			T1Q + T2Q + 30	ns	
t_{OBSH}	CP1 to output phase bank signal rising edge			46				ns	
t_{MOBSH}	MCLK to output phase bank signal rising edge	0		20				ns	
t_{IDS}	Input data setup to CP3	–3						ns	
t_{MIDS}	Input data setup to MCLK	–50			25 – T1Q – T2Q			ns	
t_{IDH}	Input data hold from CP3	28						ns	
t_{MIDH}	Input data hold from MCLK	78			T1Q + T2Q + 3			ns	
t_{ODH}	Output data hold from CP1	35		55				ns	
t_{MODH}	Output data hold from MCLK	10		25				ns	

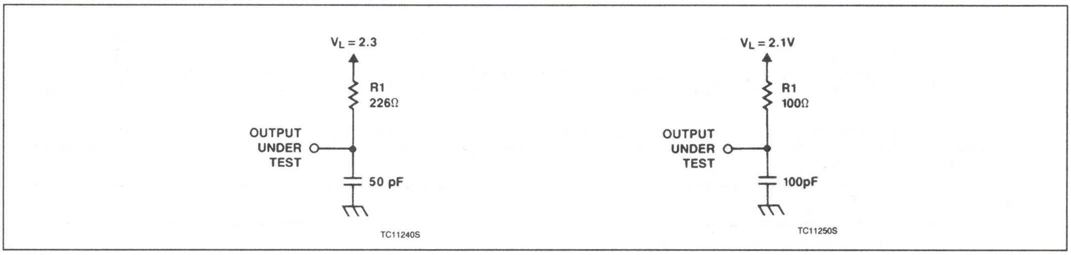
AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	150ns CYCLE			> 150ns CYCLE			UNIT	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
t_{ODS}	CP3 to output data stable			70				ns	
t_{MODS}	MCLK to output data stable			120			T1Q + T2Q + 45	ns	
t_{DI}	SC/WC rising edge to output driver turn on	18						ns	
t_{HS}	Halt setup to CP2	0						ns	
t_{MHS}	Halt setup to MCLK	−10						ns	
t_{HH}	Halt hold from CP2	50						ns	
t_{MHH}	Halt hold from MCLK	60			T1Q + 22			ns	
t_{SIS}	Status input setup to CP1	10						ns	
t_{MSIS}	Status input setup to MCLK	40						ns	
t_{SIH}	Status input hold from CP1	20						ns	
t_{MSIH}	Status input hold from MCLK	0						ns	
t_{SCRS}	SCR setup to CP1	0						ns	
t_{MSCRS}	SCR setup to MCLK	25						ns	
t_{SCRH}	SCR hold from CP2 (Slow CP2)	20						ns	See Diagram for CP2
t_{MSCRH}	SCR hold from MCLK	63			ST1Q − 12			ns	Slow: T1Q
t_{INTS}	INT setup to CP1	10						ns	
t_{MINTS}	INT setup to MCLK	40						ns	
t_{INTH}	INT hold from CP1	10						ns	
t_{MINTH}	INT hold from MCLK	−10						ns	
t_{CYU}	CP4 to CY update			60				ns	
t_{MCYU}	MCLK to CY update			−10			28 − T4Q	ns	
t_{NZU}	CP4 to NZ update			60				ns	
t_{MNZU}	MCLK to NZ update			−5			33 − T4Q	ns	
t_{IRU}	CP4 to IR update			75				ns	
t_{MIRU}	MCLK to IR update			20			58 − T4Q	ns	
t_{PSU}	CP4 to PS update			60				ns	
t_{MPSU}	MCLK to PS update			−10			28 − T4Q	ns	
t_{ACC}	Program memory access time (address stable to valid instruction)			60		T2Q + T3Q + T4Q − 52		ns	
t_{IO}	I/O port output enable time (bank signal to valid data on bus)			24			T1Q + T2Q − 51	ns	
t_{RW}	Reset pulse width	150			t_{PC}			ns	
t_{NMIW}	NMI pulse width	50						ns	
t_{NMIS}	NMI setup to CP2	15						ns	See Note 5
t_{MNMIS}	NMI setup to MCLK	10						ns	See Note 5

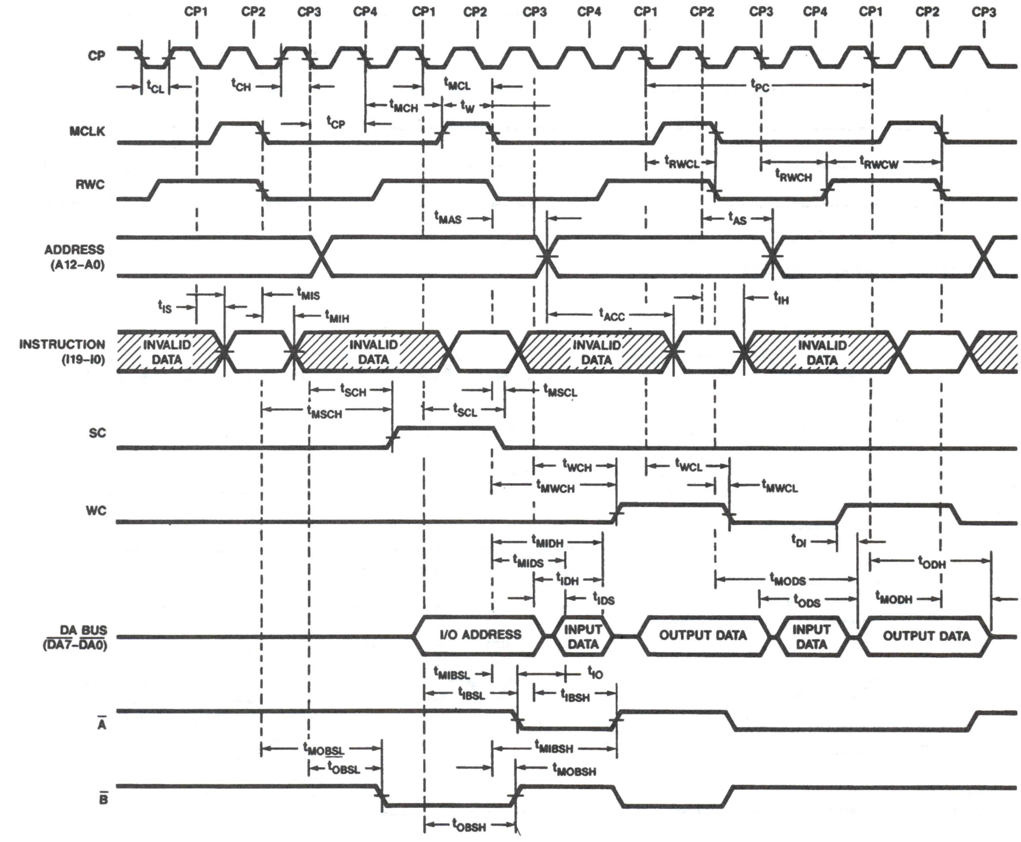
NOTES:

1. Inputs swing between 0V and 3V. All outputs are measured at 1.5V with loading as specified in the test circuits.
2. CP1, CP2, CP3, and CP4 refer to the clock pulse that causes the first, second, third, and fourth 8X401 quarter cycles, respectively. Parameters referenced to MCLK, CP1, CP2, CP3, and CP4 are measured to the falling edge of those signals. T1Q, T2Q, T3Q, and T4Q represent time intervals for the first, second, third, and fourth 8X401 quarter cycles, respectively. Duty cycle can be from 40% to 60%.
3. Instructions must be setup before CP1.
4. t_{WL} represents t_{SCL} and t_{WCL}. t_{MWL} represents t_{MSCL} and t_{MWCL}.
5. This guarantees NMI is serviced in the current cycle.

TEST CIRCUIT


V$_L$ = 2.3
R1
226Ω
OUTPUT UNDER TEST
50 pF
TC11240S
V$_L$ = 2.1V
R1
100Ω
OUTPUT UNDER TEST
100pF
TC11250S

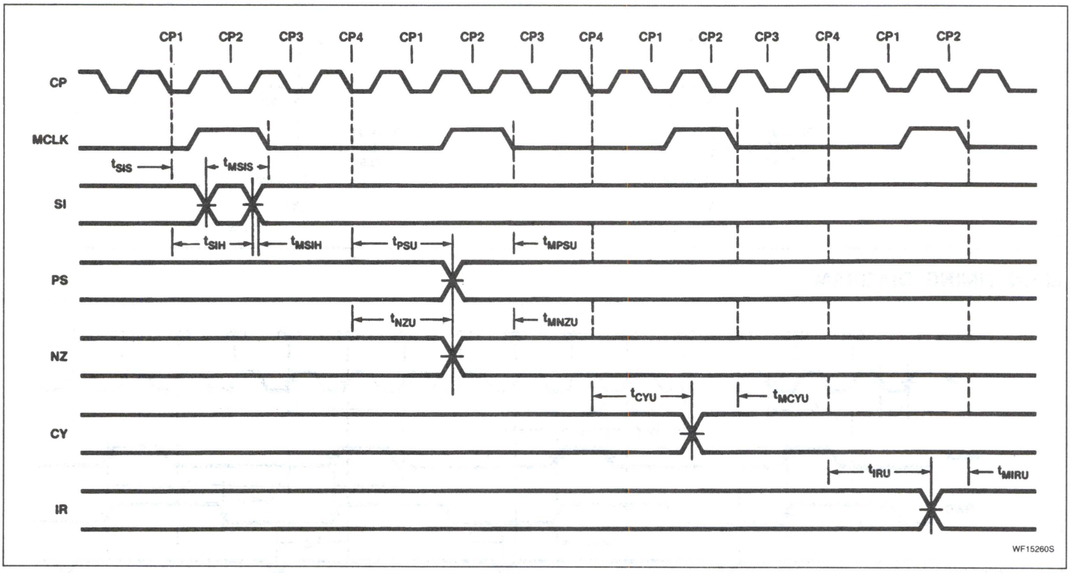
MAIN TIMING DIAGRAM


CP1
CP2
CP3
CP4
CP
MCLK
RWC
ADDRESS (A12–A0)
INSTRUCTION (I19–I0)
INVALID DATA
SC
WC
DA BUS (DA7–DA0)
I/O ADDRESS
INPUT DATA
OUTPUT DATA
A
B
t$_{CL}$
t$_{CH}$
t$_{CP}$
t$_{MCL}$
t$_{MCH}$
t$_W$
t$_{PC}$
t$_{RWCL}$
t$_{RWCH}$
t$_{RWCW}$
t$_{MAS}$
t$_{AS}$
t$_{IS}$
t$_{MIS}$
t$_{MIH}$
t$_{ACC}$
t$_{IH}$
t$_{SCH}$
t$_{MSCH}$
t$_{SCL}$
t$_{MSCL}$
t$_{WCH}$
t$_{MWCH}$
t$_{WCL}$
t$_{MWCL}$
t$_{MIDH}$
t$_{MIDS}$
t$_{IDH}$
t$_{IDS}$
t$_{DI}$
t$_{MODS}$
t$_{ODS}$
t$_{ODH}$
t$_{MODH}$
t$_{MIBSL}$
t$_{IO}$
t$_{IBSL}$
t$_{IBSH}$
t$_{MOBSL}$
t$_{OBSL}$
t$_{MIBSH}$
t$_{MOBSH}$
t$_{OBSH}$

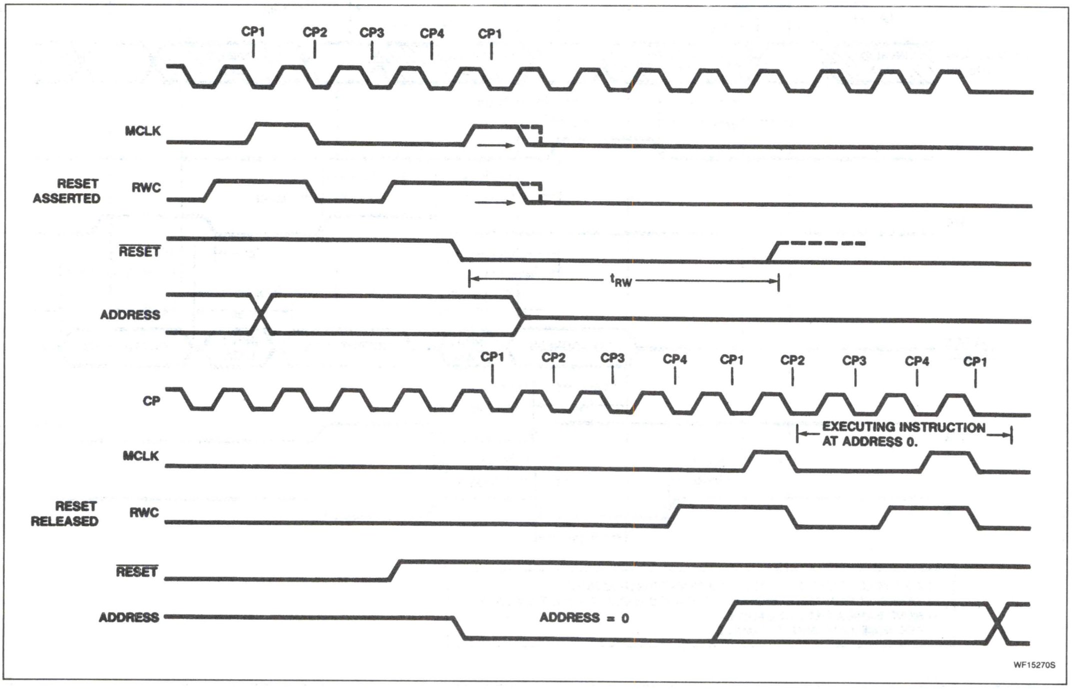
THE 3 CYCLES REPRESENT THE FOLLOWING INSTRUCTIONS:
- **TRANSMIT IMMEDIATE ADDRESS TO BANK B (SELECT A PORT ON BANK B)**
- **MOVE BANK A DATA TO BANK B**
- **ADD IMMEDIATE DATA TO BANK A**

WF15250S

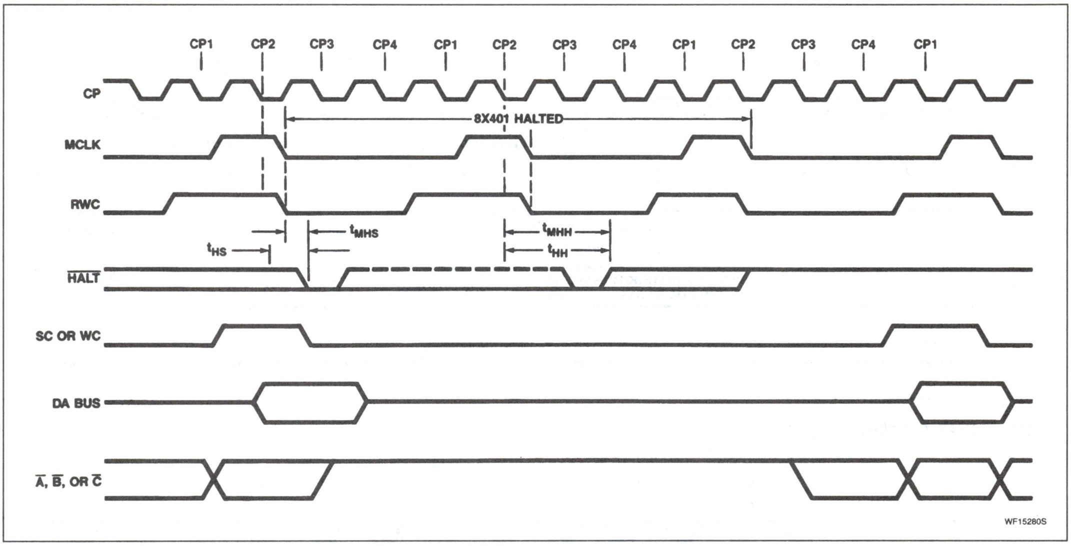
TIMING SUPPLEMENT: STATUS


CP1
CP2
CP3
CP4
CP
MCLK
SI
PS
NZ
CY
IR
t_{SIS}
t_{MSIS}
t_{SIH}
t_{MSIH}
t_{PSU}
t_{MPSU}
t_{NZU}
t_{MNZU}
t_{CYU}
t_{MCYU}
t_{IRU}
t_{MIRU}
WF15260S

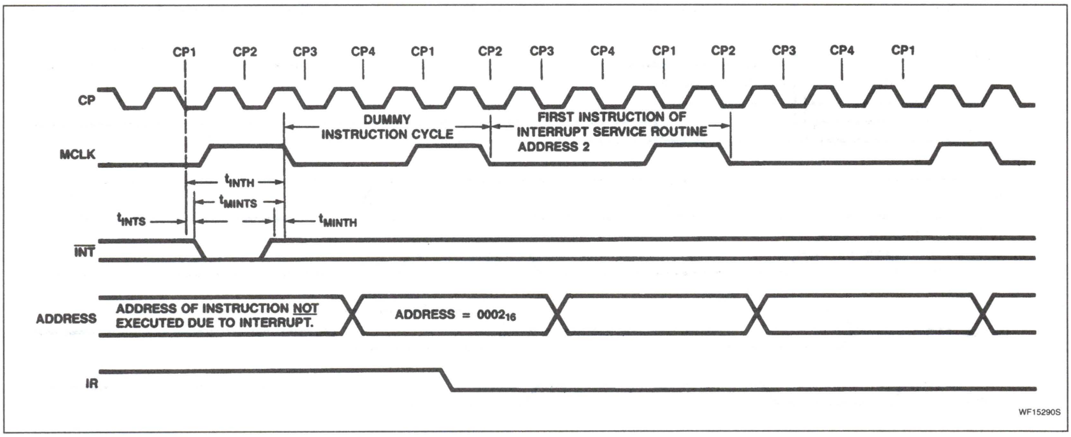
TIMING SUPPLEMENT: $\overline{\text{RESET}}$


CP1
CP2
CP3
CP4
MCLK
RESET ASSERTED
RWC
$\overline{\text{RESET}}$
ADDRESS
t_{RW}
CP
EXECUTING INSTRUCTION AT ADDRESS 0.
RESET RELEASED
ADDRESS = 0
WF15270S

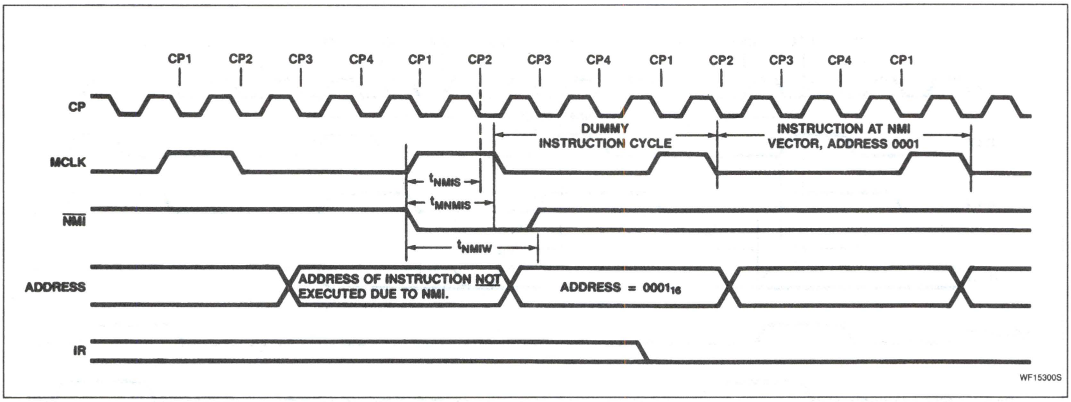
TIMING SUPPLEMENT: $\overline{\text{HALT}}$


CP1
CP2
CP3
CP4
CP1
CP2
CP3
CP4
CP1
CP2
CP3
CP4
CP1
CP
8X401 HALTED
MCLK
RWC
t_{MHS}
t_{HS}
t_{MHH}
t_{HH}
$\overline{\text{HALT}}$
SC OR WC
DA BUS
$\overline{\text{A}}$, $\overline{\text{B}}$, OR $\overline{\text{C}}$
WF15280S

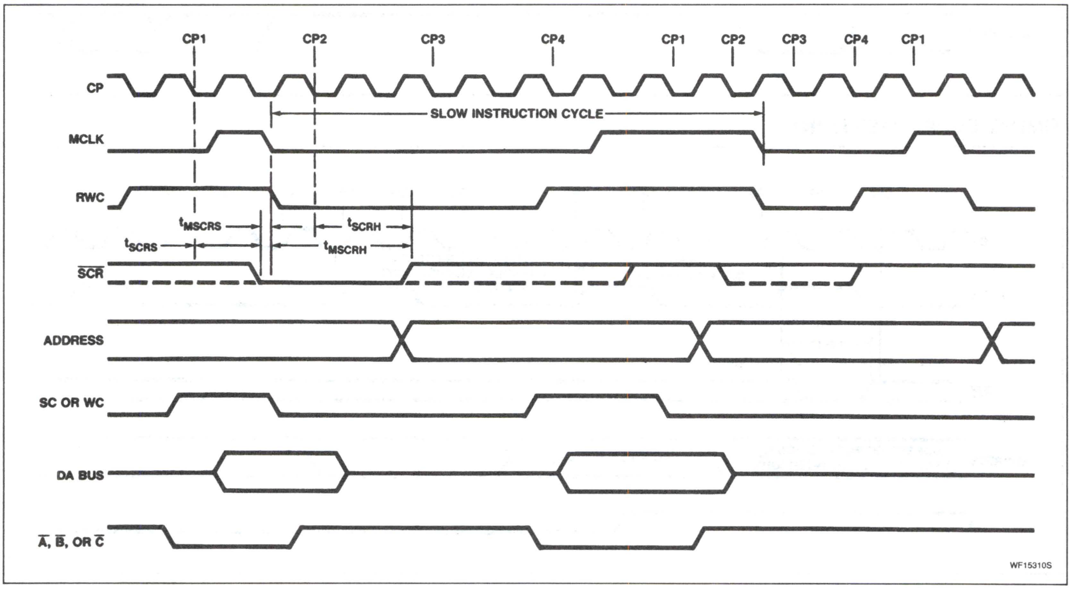
TIMING SUPPLEMENT: $\overline{\text{INT}}$


CP1
CP2
CP3
CP4
CP1
CP2
CP3
CP4
CP1
CP2
CP3
CP4
CP1
CP
DUMMY INSTRUCTION CYCLE
FIRST INSTRUCTION OF INTERRUPT SERVICE ROUTINE ADDRESS 2
MCLK
t_{INTH}
t_{MINTS}
t_{INTS}
t_{MINTH}
$\overline{\text{INT}}$
ADDRESS
ADDRESS OF INSTRUCTION NOT EXECUTED DUE TO INTERRUPT.
ADDRESS = 0002_{16}
IR
WF15290S

TIMING SUPPLEMENT: $\overline{\text{NMI}}$


CP1
CP2
CP3
CP4
CP1
CP2
CP3
CP4
CP1
CP2
CP3
CP4
CP1
CP
MCLK
NMI
ADDRESS
IR
DUMMY INSTRUCTION CYCLE
INSTRUCTION AT NMI VECTOR, ADDRESS 0001
tNMIS
tMNNMIS
tNMIW
ADDRESS OF INSTRUCTION NOT EXECUTED DUE TO NMI.
ADDRESS = 0001₁₆
WF15300S

TIMING SUPPLEMENT: $\overline{\text{SCR}}$


CP1
CP2
CP3
CP4
CP1
CP2
CP3
CP4
CP1
CP
SLOW INSTRUCTION CYCLE
MCLK
RWC
tMSCRS
tSCRH
tSCRS
tMSCRH
SCR
ADDRESS
SC OR WC
DA BUS
A, B, OR C
WF15310S

Signetics

Microprocessor Products

AN4011
An 8X401-Based Raster Graphics System

Application Note

Computer graphic displays have gained popularity over the last few years as man-machine interfaces. The major display technologies are storage tube, stroke/vector/calligraphic, and raster scan displays.

Of these, raster scan displays are by far the most popular. Two characterizing features of raster scan displays are: the requirement of large amounts of memory for display purposes, and the need to execute relatively large numbers of instructions for simple display operations. Raster scan graphics can thus be described as high-speed bit manipulations of large amounts of data.

In the past, the above-mentioned memory and execution characteristics severely limited the popularity of raster scan displays. However, as memory densities and processing powers increased and as costs in general fell, raster scan displays grew in popularity. Lower costs and increased processing powers also led to the distribution of processing in raster graphics systems. Display processors which could offload much of the tedious bit-level manipulations from a host became common. These display processors provided a high-level graphics software interface to the host, improving system operation.

The Signetics 8X401 Microcontroller continues this trend of increased powers at lower costs. With its powerful bit-manipulation orientated instruction set, its high-speed, and extensive peripheral family support, the 8X401 is ideally suited for use as a high-performance display processor in raster scan graphics systems.

RASTER GRAPHICS DISPLAYS

In raster graphics, the Cathode Ray Tube (CRT) is the basic display element. Imagine the CRT screen overlaid with an x-y grid. At each intersection there exists a picture element or pixel. Each pixel is comprised of red, green and blue phosphors (refer to Figure 1). These phosphors, when hit by an electron beam, emit light. By appropriately mixing these three primary colors, a pixel can be defined to be any one of a variety of colors.

The light emitted by the phosphors decays with time. Eventually no light will be emitted and the screen display will be lost. To maintain a stable picture display, the phosphors must be periodically re-excited or refreshed. Generally if phosphor refresh is not done at least 40 times a second (40Hz) the screen will appear to flicker.

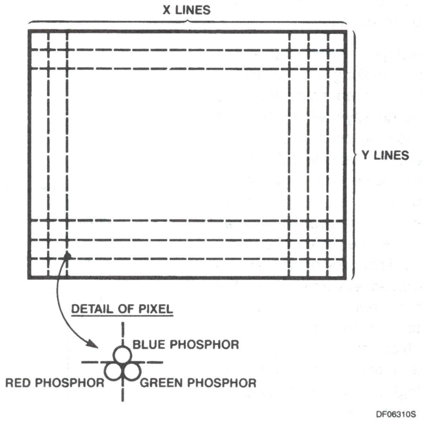


Figure 1. CRT Screen With Pixel Detail

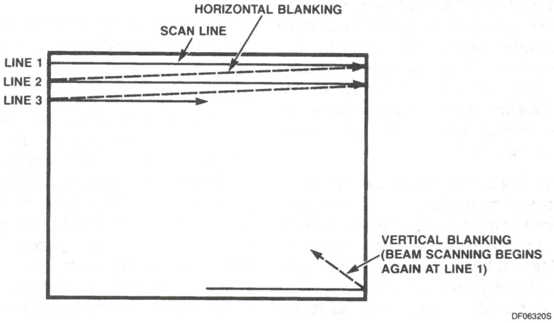


Figure 2. Non-Interlaced Displays

Screen refresh occurs along a series of horizontal scan lines. Three electron beams (one for each phosphor) start from the top lefthand corner of the CRT screen and are scanned horizontally along a line from the left to the right side of the screen. Upon reaching the right side of the screen the beams are turned off (or horizontally blanked) to prevent spurious images from being drawn. They are then sent back and down to the left side of the screen to refresh the next line to be scanned. (The next line to be scanned is dependent upon whether an interlaced or non-interlaced display is used. Please see page two for a complete description of interlaced and non-interlaced displays.)

The beam return time from the right to the left of the screen is called flyback or retrace. When the last line is scanned (again dependent upon interlaced vs. non-interlaced), the beams are again turned off (or vertically blanked), but now returned to the top lefthand corner of the screen. The screen refresh process is then started anew.

There are two basic methods of screen refresh: interlaced displays and non-interlaced displays. The difference between the two methods can be easily and accurately explained as follows: First, assign each scan line on the CRT screen a sequential number (starting with number 1 for the first line). Then define an odd field to consist of all the odd numbered lines and an even field to consist of all the even numbered lines. One entire CRT screen then consists of the odd field intermixed with the even field.

Non-interlaced displays refresh the entire screen sequentially, making no distinction between the odd and even fields (refer to Figure 2). Interlaced displays refresh the odd field first, then refresh the even field (refer to Figure 3). Non-interlaced displays refresh the entire screen in the same time required for interlaced displays to refresh just one field. Interlaced displays therefore require twice as long to refresh the entire screen as non-interlaced systems. This slower screen refresh rate for interlaced systems allows the use of lower performance components; reducing system costs. However, with high-performance graphics systems, interlaced displays may cause noticeable screen flicker. Therefore most high-performance systems use non-interlaced displays.

Screen refreshing is done using pixel information stored in the frame buffer. An entry in the frame buffer contains information on its corresponding pixel. As a bare minimum, the frame buffer must contain as many entries as there are displayable pixels. For example, if a system contains 1M pixels, then the display memory must contain a minimum of 1M entries. (Display memories are often larger than the minimum size to allow for different fonts, patterns, etc.)

Each entry in the frame buffer can consist of multiple bits of information. For example, four bits per entry in a color system would allow 16 colors ($2^4 = 16$) to be simultaneously displayed (Many CAD/CAM systems use 8 bits per pixel allowing 256 colors to be simultaneously displayed). Without any additional hardware support, the palette for such a 4-bit per pixel color system is 16 colors. If this palette is not sufficient, a color lookup table can be used to enlarge the spectrum of displayable colors.

Resolutions of raster graphics systems are specified as the number of horizontal pixels by the number of vertical pixels. The higher the resolution, the better the picture quality (typical high resolution displays, such as the ones in CAD/CAM systems, run from about 1024 × 1024 pixels on up). But as screen resolutions increase, the number of instructions needed to draw pictures also increases. Consequently a picture may require a relatively long time to be drawn. Generally the faster a picture can be drawn, the better the system is. The tradeoff for higher resolutions and faster system operations is, obviously, system cost.

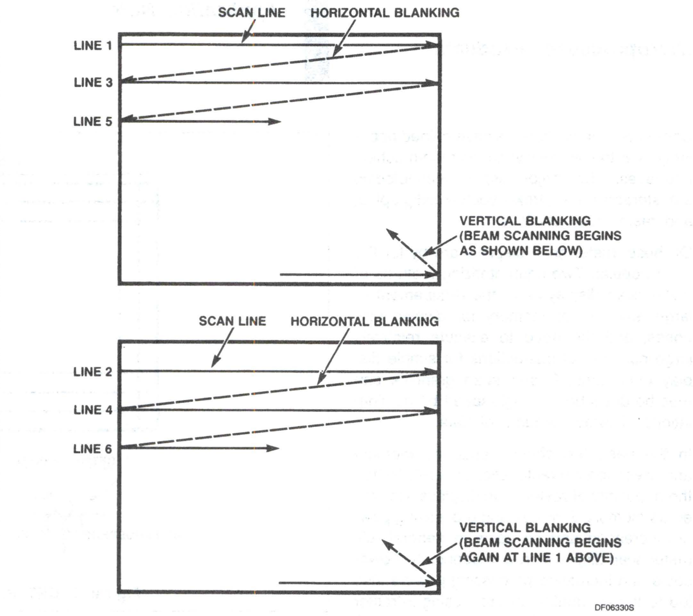

SCAN LINE
HORIZONTAL BLANKING
LINE 1
LINE 3
LINE 5
VERTICAL BLANKING (BEAM SCANNING BEGINS AS SHOWN BELOW)
SCAN LINE
HORIZONTAL BLANKING
LINE 2
LINE 4
LINE 6
VERTICAL BLANKING (BEAM SCANNING BEGINS AGAIN AT LINE 1 ABOVE)
DF06330S

Figure 3. Interlaced Displays

When multiple bits per pixel are combined with higher resolutions, the demands for memory can grow geometrically. For example, with a 1024 × 1024 display, information in 1M pixels must be stored. With a simple 4-plane system, 4M bits or 512K bytes of memory are needed just for display purposes. Efficient memory management becomes a major task of any raster graphics system. With large amounts of data being continuously transferred to the screen, a small inefficiency in memory access can multiply into a severe limitation. The memory management task is complicated by the multiple demands upon the display memory.

These multiple demands are made by the update process and the display process. The update process must access the display memory to update the picture, and the display process must access the display memory to refresh the screen. The display process, however, has more time-critical needs than the update process, and should have priority access to the display memory. If the display process cannot get timely access to the display memory, distorted pictures may result.

For an example of the display process time critical needs, take the case of a 1024 × 1024 non-interleaved display refreshed at 60Hz. Retrace times are roughly 25% of the time required to display a frame. Define the pixel rate as the speed with which pixels must be sent to the screen. The pixel rate in the system above is determined by:

$$\begin{aligned}\text{pixel rate} &= (\text{number of pixels})/\\ &\quad(\text{frame time} - \text{retrace time})\\ &= (1024 \times 1024)/\\ &\quad<(1/60) - .25(1/60)>\\ &= 83.9\text{MHz}\end{aligned}$$

[For the sake of comparison, the pixel rate for a comparable interlaced system is:

$$\begin{aligned}\text{pixel rate} &= (\text{number of pixels}/2)/\\ &\quad(\text{field time} - \text{retrace time})\\ &= <(1024 \times 1024)/2>/\\ &\quad<(1/60) - .25(1/60)>\\ &= 41.5\text{MHz}]\end{aligned}$$

Display information on a pixel must be acquired from the display memory and sent to the screen every 12-ns (1/83.9 MHz = 12-ns). Traditionally, these pixel rates have been met by page-mode accessing the display DRAMs during the screen refresh times. Only during the retrace or flyback times (roughly 25% of the frame time) has the update process been able to access the display DRAMs for picture update. For the remaining 75% of the frame display time, the update process is required to wait. A variety of different architectures, such as pixel caches, multiple frame buffers and segmented memory organization, attempt to overcome this page-mode access limitation.

A number of manufacturers have recently introduced dual port video DRAMs which help to alleviate the display bottleneck. These devices allow a row of the DRAM to be sent to an on-board shift register. The display process can then access the shift register to refresh the screen, while the update process can access the DRAM to draw a picture. The 75% wait requirement imposed above is reduced. The designer's attention can be turned to improving the performance of the display processor (the heart of the system).

Raster graphics systems must execute large numbers of instructions to generate even simple displays. This is due to the large number of pixels that must be specified for every picture. For example, a simple routine to clear the screen on an 8X401-based raster scan system requires the execution of almost 200,000 instructions. Changing a display that contains windows of different text might require the update of hundreds of thousands of pixels and involve millions of instructions. Pictures that require many instructions to draw may take considerable time to complete. Users have grown to expect fast displays and are annoyed by slow ones. This need for fast screen dynamics puts a premium on speeds of display processors.

Raw speed by itself, however, may not be sufficient. If a fast display processor's instruction set is not efficient at the bit manipulations required in raster scan graphics, screen dynamics may be poor. A slower processor with a highly efficient instruction set may allow screen dynamics that surpass those of a fast processor. For example, consider a byte packed with two 4-bit pixels. Suppose in this packed byte that only the pixel in the high-order bits is to be manipulated. Traditional display processors, which can only handle bit, byte or word data sizes, would require multiple instructions to operate on the high-order pixels. A sub-field orientated processor could work on a 4-bit data size, thus reducing the number of instructions needed for pixel manipulation. The total time to execute the routine may be the same, but in terms of cost, the slower processor system with its more powerful instruction set may be a much better choice.

Designers of high-performance display systems have typically used bit-slice processors to supply the required processor speeds. Their flexible architectures allow them to be used in a variety of configurations. Excellent screen dynamics are possible with these processors.

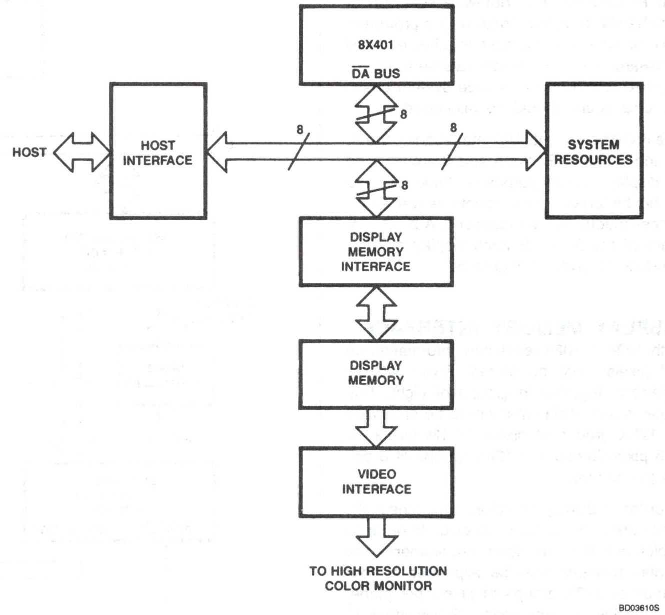


Figure 4. Raster Graphics System Block Diagram

However, bit-slice processor designs lead to increased chip counts, greater power consumption, greater cost and reduced reliability. They do not have any sub-field manipulation capability and their I/O capabilities are limited. Their building block architecture leads to greatly expanded microinstruction words. Pipelining techniques are generally used, leading to a penalty for branching. However, due to the limited number of alternatives available, bit-slice designs are used in high-performance systems despite their liabilities.

Single chip video processors offer more instruction set functionality but at slower speeds than bit-slice processors. Video processors, in general, provide a high-level software interface to a host processor, and generate DRAM timing and control signals. They refresh the CRT screen by accessing the display DRAMs using page-mode access. For medium resolution applications, these video processor chips provide a relatively efficient single chip graphics controller solution. Systems using video processors generally use less chips, consume less power and cost less than bit-slice designs.

For high-resolution displays, such as those in CAD/CAM systems, however, the screen dynamics available with video processors may be unacceptable. Their architectures are fixed in silicon and alternate designs (such as systems using the video DRAMs) cannot easily be implemented. These chips typically can update one pixel a bit at a time. For multiplaner displays, this may cause unacceptably slow pixel drawing times. Some systems use multiple video chips (one for each plane) to overcome this single bit drawing limitation.

The 8X401, used as a display processor, combines the high-speed and flexibility of bit-slice designs with the powerful instruction set, reduced chip count, and lower cost of video processor designs. The high-performance available with an 8X401 based design makes it a very attractive alternative to both bit-slice and video processor based graphics system designs. A block diagram of a raster scan graphics system using the 8X401 as the display processor is shown in Figure 4.

8X401 SYSTEM DESIGN

Requirements for the system in Figure 4 are 1024 × 1024 resolution with 60Hz non-interlaced display, with the capability to simultaneously display 16 colors out of a total palette of 4096 colors. The 8X401 should perform all tasks required to refresh the screen and should also provide a high-level software interface to a 16-bit host (a 68010 Microprocessor), executing such commands as DRAW

LINE, BIT-BLOCK TRANSFER (BIT-BLT), and POLYGON FILL. Software selection of peripherals to reduce costs will be preferred; only for time critical access requirements will hardware selects of peripherals be used. To improve reliability and reduce system costs, the chip count should be minimized.

The 8X401 segments I/O into 3 banks: the A, B, and C banks. Banks A and B are be used for display memory purposes. Bank C is used for host interface requirements as well as for miscellaneous system functions. A block diagram of the A and B bank display memory interface is given in Figure 5.

DISPLAY MEMORY INTERFACE

With 1024 × 1024 resolution, information on 1M pixels must be stored. Pixels will be organized together in groups of eight. This requires the display memory to be organized as 128K groups of pixels (< 1M pixels > / < 8 pixels/group > = 128K groups of 8 pixels per group).

In order to display 16 colors simultaneously, 4-bits are required for each pixel. In order to implement this 16 color requirement, the display memory must be expanded to four planes of 128K groups of pixels per plane. This display memory organization is shown in Figure 6. Physically, the display memory is implemented as four planes of 128K bytes of Texas Instruments 4161 dual-port video DRAMs per plane, and two Intel 8203 DRAM controllers. These controllers help simplify system design by providing all timing and refresh signals required by the DRAMs, thereby offering a transparent DRAM interface to the 8X401.

The 4161 DRAMs used in the display memory have 200-ns access time specifications (300-ns cycle time). Upon receiving a memory access request, the 8203 requires about 100-ns before initiating a DRAM access cycle. The DRAM controller-DRAM access path takes 300-ns to complete, with cycle times totaling 400-ns. With successive memory accesses, the 100-ns precharge time of the current access can be overlapped with the 100-ns DRAM controller propagation delay of the next cycle. This allows accesses on the display memory to be cycled through every 300-ns.

Since the access path of the memory is relatively slow compared to the 8X401 instruction cycle time, it forces the 8X401 to wait for the completion of the memory cycle.

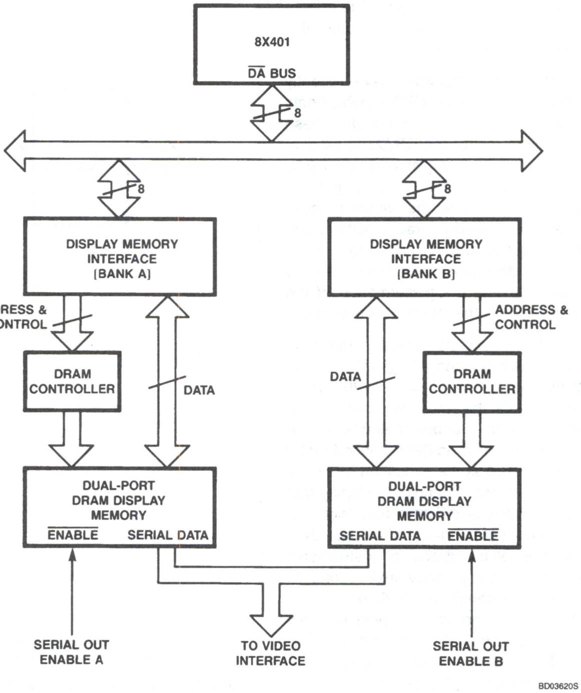

8X401
$\overline{DA}$ BUS
8
8
8
DISPLAY MEMORY INTERFACE (BANK A)
DISPLAY MEMORY INTERFACE (BANK B)
ADDRESS & CONTROL
ADDRESS & CONTROL
DRAM CONTROLLER
DATA
DATA
DRAM CONTROLLER
DUAL-PORT DRAM DISPLAY MEMORY
$\overline{ENABLE}$ SERIAL DATA
DUAL-PORT DRAM DISPLAY MEMORY
SERIAL DATA $\overline{ENABLE}$
SERIAL OUT ENABLE A
TO VIDEO INTERFACE
SERIAL OUT ENABLE B
BD03620S

Figure 5. 8X401 – Display Memory Interface Block Diagram

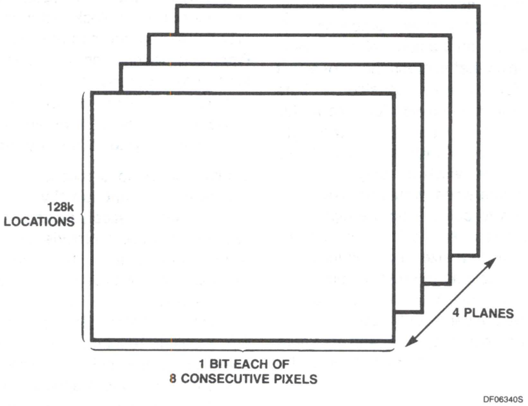

128k LOCATIONS
4 PLANES
1 BIT EACH OF 8 CONSECUTIVE PIXELS
DF06340S

Figure 6. Complete Display Memory Organization

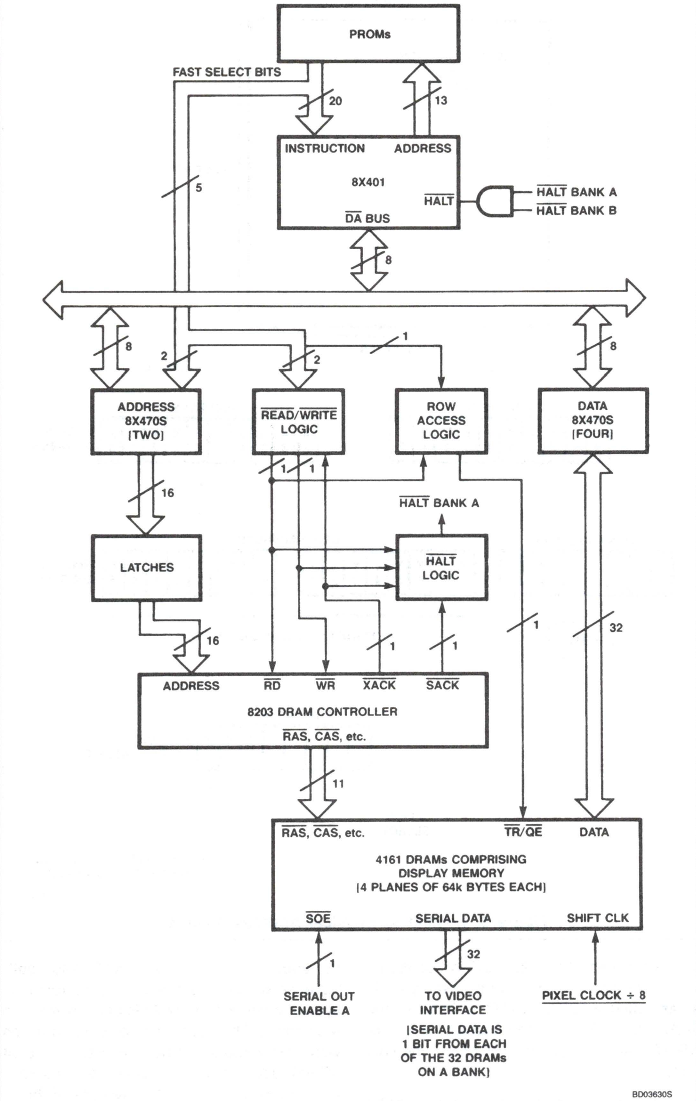

PROMs
FAST SELECT BITS
20
13
INSTRUCTION
ADDRESS
8X401
HALT
DA BUS
HALT BANK A
HALT BANK B
5
8
8
2
2
1
8
ADDRESS 8X470S (TWO)
READ/WRITE LOGIC
ROW ACCESS LOGIC
DATA 8X470S (FOUR)
16
1
1
HALT BANK A
LATCHES
HALT LOGIC
16
1
1
1
32
ADDRESS
RD
WR
XACK
SACK
8203 DRAM CONTROLLER
RAS, CAS, etc.
11
RAS, CAS, etc.
TR/QE
DATA
4161 DRAMs COMPRISING DISPLAY MEMORY (4 PLANES OF 64k BYTES EACH)
SOE
SERIAL DATA
SHIFT CLK
1
32
SERIAL OUT ENABLE A
TO VIDEO INTERFACE
(SERIAL DATA IS 1 BIT FROM EACH OF THE 32 DRAMs ON A BANK)
PIXEL CLOCK ÷ 8
BD03630S

Figure 7. Detail of 8X401 – Display Memory Interface [All 8X470s Above Mapped Onto the 8X401's A Bank]

For optimal performance, the 8X401 should initiate a memory request and then proceed with another task. It can return two instructions or 300-ns later and the memory access should be completed.

However if the 8X401's memory request was delayed due to a conflict with a DRAM refresh, then the 300-ns access time is not valid. If the 8X401 initiates an access and returns 300-ns later, the memory cycle may not have been completed. To ensure proper operation in the case of delayed access, the DRAM controller's acknowledge signal (SACK*) is used.

SACK* is asserted by the 8203 within 150-ns of a memory access request if the request is granted immediately. If not granted immediately, SACK* will not be asserted within 150-ns. The lack of SACK* being asserted within 150-ns causes the HALT circuitry of Figure 7 to halt the 8X401, effectively causing a wait state. When the 8203 cycle complete signal (XACK*) is asserted, the 8X401 is revived, thereby ensuring proper operation.

As is apparent from Figure 7, the HALT* input to the 8X401 is formed by ANDing the HALT* signals from bank A and bank B. These two HALT* signals exist due to the fact that the system shown in Figure 5 segments the display memory. Four planes of 64K bytes (and an associated 8203 controller) are placed on bank A and four planes of 64K bytes (and its 8203 controller) on bank B. Full advantage can be taken of the bank-to-bank transfer capabilities and the dual bank output capabilities of the 8X401. Though segmenting the display memory introduces a certain duplication of effort, it will be apparent that the improved screen dynamics available with such an organization is well worth the additional cost.

In order to address the 64K bytes of DRAM on each bank, 16-bit addresses are needed. Two Signetics 8X470 8-bit I/O ports on each bank are used as address registers to generate 16-bit addresses. (Figure 7 shows the address 8X470s). The address 8X470s (ADDR HIGH and ADDR LOW) are selected using fast select bits. The expenditure of two instructions per bank to preselect them (one for each 8X470) is not needed. In many tight loops, the additional preselect overhead would be an unnecessary burden, severely limiting screen dynamics. The addition of four fast select bits to select the four address registers in the system is therefore a wise investment. This will be apparent from the code segments given in the software section on page 10.

An 8X401-Based Raster Graphics System

AN4011

Pixel addressing on the display memory is done using the physical addresses contained in the address 8X470s. However, the host address pixels according to their logical X-Y (horizontal-vertical) addresses. Address mapping must be done on the host's logical addresses by the 8X401. With a 1024 × 1024 resolution, 10-bits are used by the host for each of the X and Y coordinates. The 8X401 translates the X and Y coordinates into a 1-bit bank select, a 16-bit byte address and a 3-bit pixel address. The translation done by the 8X401 on the logical addresses is shown in Figure 8.

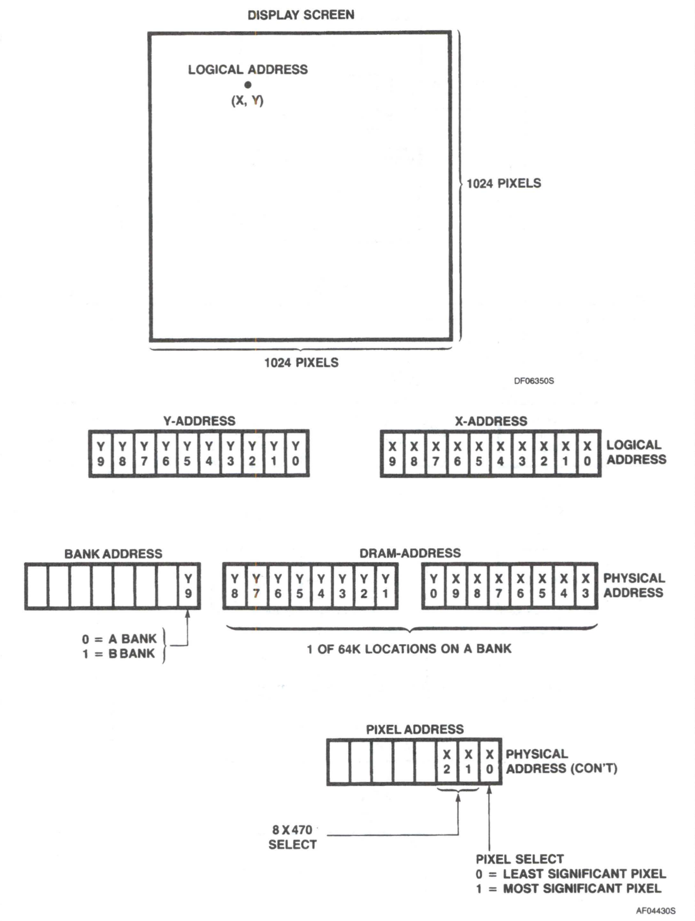

DISPLAY SCREEN
LOGICAL ADDRESS
(X, Y)
1024 PIXELS
1024 PIXELS
DF06350S
Y-ADDRESS
Y 9 | Y 8 | Y 7 | Y 6 | Y 5 | Y 4 | Y 3 | Y 2 | Y 1 | Y 0
X-ADDRESS
X 9 | X 8 | X 7 | X 6 | X 5 | X 4 | X 3 | X 2 | X 1 | X 0
LOGICAL ADDRESS
BANK ADDRESS
Y 9
DRAM-ADDRESS
Y 8 | Y 7 | Y 6 | Y 5 | Y 4 | Y 3 | Y 2 | Y 1
Y 0 | X 9 | X 8 | X 7 | X 6 | X 5 | X 4 | X 3
PHYSICAL ADDRESS
0 = A BANK
1 = B BANK
1 OF 64K LOCATIONS ON A BANK
PIXEL ADDRESS
X 2 | X 1 | X 0
PHYSICAL ADDRESS (CON'T)
8 X 470 SELECT
PIXEL SELECT
0 = LEAST SIGNIFICANT PIXEL
1 = MOST SIGNIFICANT PIXEL
AF04430S

Figure 8. Logical to Physical Address Mapping

A design limitation imposed by DRAM controllers is that addresses input to them do not change while a memory access cycle is in progress. As mentioned previously, the 8X401 can execute two instructions while a display memory access cycle is underway. For optimal system performance, it is desirable in some instances to allow the 8X401 to expend these two instructions on updating the addresses contained in the address 8X470s. This would allow another memory access to begin immediately after the current one ends. A solution must be found to allow addresses in the address 8X470s to be changed without violating the DRAM controller design limitation mentioned above.

The solution used in the system of Figure 7 is to latch the outputs of the address 8X470s using 74F374 octal latches. The access command signals input to the 8203 DRAM controllers (READ* and WRITE*) are used as the gating signals for the latches. Whenever both READ* and WRITE* are high, the latches are transparent. However, with the falling edge of either READ* or WRITE*, the latches close. The outputs of the latches are then input as stable memory addresses to the DRAM controllers, ensuring that the design limitation specified above is met.

The 8203 DRAM controllers initiate a memory access cycle upon the assertion of either the READ* or WRITE* inputs. However, these inputs, expected by the 8203 controllers, introduce another design problem — the 8X401 does not have the traditional READ/WRITE* signal that Microprocessors do. Either an 8X470 port or fast select bits must be used to generate the READ* and WRITE* inputs to the DRAM controllers. For the sake of improved screen dynamics, fast select bits are used for each bank to generate the READ* and WRITE* commands. These fast select bits are latched by the READ*/WRITE* logic shown in Figure 7. The outputs of these latches are fed to the DRAM controllers and to the address latches.

Once either READ* or WRITE* is asserted by the READ*/WRITE* logic, it will remain asserted until the display memory access cycle has completed. The 8203 DRAM controller informs of the completion of a memory cycle by asserting XACK*. XACK* is input to the READ*/WRITE* logic to deassert READ* or WRITE* with the completion of the memory cycle.

Any access to the display memory will not be on only a single plane. The same location in all four planes on a bank will be simultaneously accessed. A byte from each plane is accessed, and with four planes a total of 32 bits are then transferred. Four 8X470s (shown Figure 7) are used as data ports to hold the 32 bits of data transferred between the 8X401 and the display memory.

The time-critical access needs of the data 8X470s are not nearly as great as those of the address 8X470s. So normal 8X401 software select procedures are used, reducing system costs. Each 8X470 is packed with two 4-bit pixels. The host's X-Y logical addresses were translated into a bank address, a 16-bit DRAM address and a 3-bit pixel address (refer to Figure 8). The top two bits of the 3-bit pixel address select one of the four 8X470s on a bank. The least significant bit selects one of the two pixels within the selected 8X470.

As mentioned above, the data 8X470s were packed with two 4-bit pixels. However this data organization is markedly different from the display memory data organization. Whenever an access is made from the display memory, a byte from each plane is accessed.

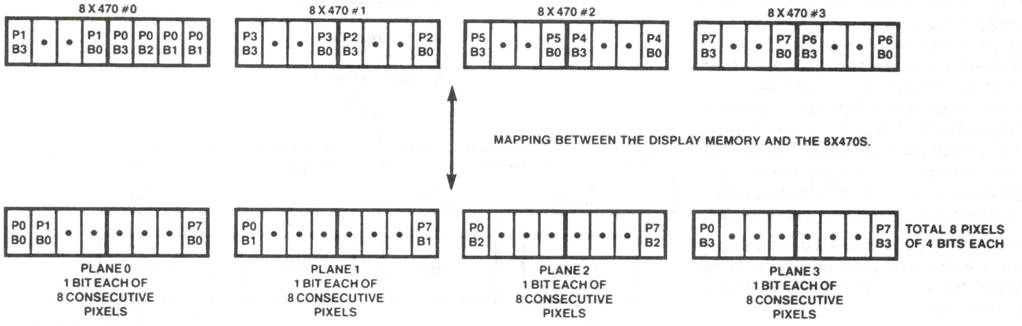

8 X 470 #0
P1 B3
P1 B0
P0 B3
P0 B2
P0 B1
P0 B1
8 X 470 #1
P3 B3
P3 B0
P2 B3
P2 B0
8 X 470 #2
P5 B3
P5 B0
P4 B3
P4 B0
8 X 470 #3
P7 B3
P7 B0
P6 B3
P6 B0
MAPPING BETWEEN THE DISPLAY MEMORY AND THE 8X470S.
P0 B0
P1 B0
P7 B0
P0 B1
P7 B1
P0 B2
P7 B2
P0 B3
P7 B3
TOTAL 8 PIXELS OF 4 BITS EACH
PLANE 0 1 BIT EACH OF 8 CONSECUTIVE PIXELS
PLANE 1 1 BIT EACH OF 8 CONSECUTIVE PIXELS
PLANE 2 1 BIT EACH OF 8 CONSECUTIVE PIXELS
PLANE 3 1 BIT EACH OF 8 CONSECUTIVE PIXELS
AF04440S

Figure 9. Display Memory to 8X470 Mapping

Each byte of the display memory plane contains one bit each of eight consecutive pixels. The four planes in parallel combine to form the 4-bits that define each pixel. This reformatting of data between the data 8X470s and the display memory (shown in Figure 9) is a matter of appropriate cross wiring between the 8X470s and the display memory. No additional circuitry is needed to reformat the data.

DISPLAY CIRCUITRY

Dual port DRAMs are used to implement the display memory. To differentiate between row access and random access, the DRAMs have an input named TR*/QE*. This input is also used to enable data output for random access READ cycles. One fast select bit is used to assert TR*/QE* for row accesses. A row access is initiated by the assertion of the TR*/QE* and READ (for the appropriate bank) fast select bits. The row access circuitry of Figure 7 takes as inputs the TR*/QE* fast select bit, and outputs the DRAM TR*/QE* signal (It will also assert the DRAM TR*/QE* signal for output enables of READ cycles.).

Row accesses are required only once per raster line. With the system as previously defined, a raster line is scanned every 15.4-μs. The 8X401 is required to make a row access of the display memory every 15.4-μs. The 8X401 is interrupted once every 15.4-μs by the interrupt circuitry in Figure 10 to initiate the screen display process.

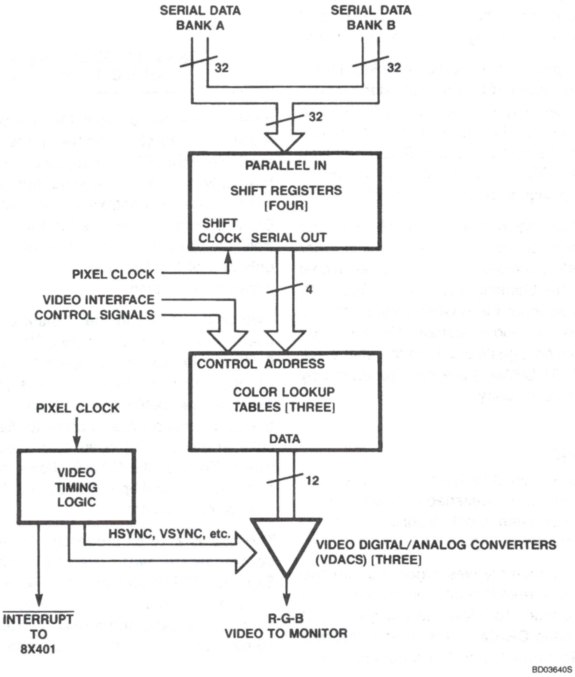

SERIAL DATA BANK A
SERIAL DATA BANK B
32
32
32
PARALLEL IN
SHIFT REGISTERS (FOUR)
SHIFT CLOCK
SERIAL OUT
PIXEL CLOCK
4
VIDEO INTERFACE CONTROL SIGNALS
CONTROL
ADDRESS
COLOR LOOKUP TABLES (THREE)
DATA
PIXEL CLOCK
VIDEO TIMING LOGIC
12
HSYNC, VSYNC, etc.
VIDEO DIGITAL/ANALOG CONVERTERS (VDACS) (THREE)
INTERRUPT TO 8X401
R-G-B VIDEO TO MONITOR
BD03640S

Figure 10. Block Diagram of Video Interface

The major responsibility of the interrupt handler is to make a row access upon the display memory. Once a row has been sent to the on-board shift register of the 4161s, it can be serially shifted out to refresh the screen. Each shift of the DRAMs provides eight new bits of information per plane. Each plane has an 8-bit high-speed shift register (shown in Figure 10) into which the serial output data from the 4161s is loaded. The display memory was organized into two banks of four planes each. The screen is refreshed from only one bank.

The bank select logic discussed in the C bank section below will allow one bank to be selected for screen refresh, while 3-stating the other bank.

The four high-speed shift registers can be shared between the eight display memory planes in the system. These shift registers are clocked at 83MHz to meet the screen rate of 12-ns pixel. The four bits of information shifted out from the four high-speed shift registers combine together to form a pixel.

The 4-bit pixel is sent to color lookup tables (shown in Figure 10) as a 4-bit address (The lookup tables expand the palette from 16 to 4096 colors.). The outputs of the lookup tables are then fed to video digital-to-analog converters (VDACs) that form the analog R-G-B video signal input to the CRT.

The VDACs require that display blanking and synchronization signals be input to it in order to properly generate the R-G-B video signal output. The blanking circuitry in Figure 10 takes as an input the 83MHz pixel clock and generates the above signals. Various other system timing signals (such as the shift clock for the 4161 DRAMs) are also generated by the blanking circuitry.

C BANK

The C bank is used for host interface requirements and for miscellaneous system functions. A complete block diagram of the C bank is shown in Figure 11.

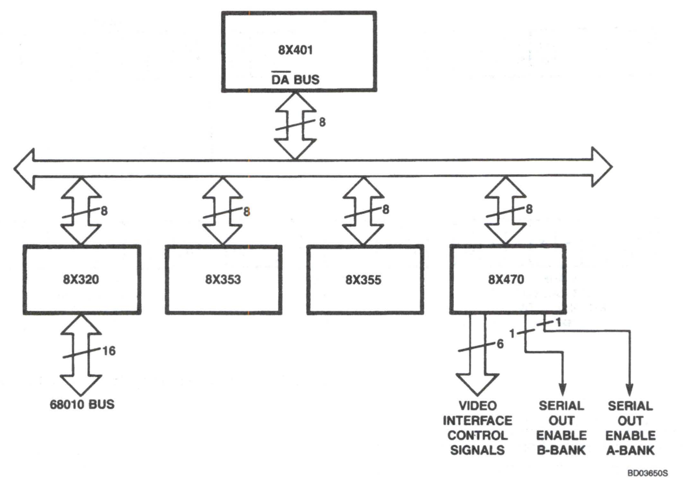


Figure 11. Block Diagram of Host Interface and System Resources [All I/O Devices Above Mapped Onto the 8X401'S C Bank]

The display memory was organized into two banks. The screen is refreshed from only one bank at a time. To allow the multiplexing of multiple video DRAMs, each 4161 includes a serial output enable pin. This input enables or disables the output from the on-board shift register. The 8X401 uses two of the eight bits available in the 8X470 mapped on the C-bank to enable or disable the serial outputs of the 4161s. One bit controls the outputs of the DRAMs on the A-bank, while the other controls the B-bank. To prevent data collision, only one of the two output control signals is asserted at a time.

Two of the 8 bits in the C-bank 8X470 are used as serial output enables. The remaining 6 bits in the 8X470 are used to implement various video interface control signals. For example, the 8X401 has to be able to access the color lookup table in order to display any color in the defined 4096 color palette. The video 8X470 on the C-bank allows the 8X401 to access the lookup table to meet the above display requirement.

The host is a 16-bit processor. To allow it to communicate with the 8-bit 8X401, a Signetics 8X320 Bus Interface Register Array is used. This chip contains an 8X401 interface on one side, and a user selectable 8 or 16-bit interface on the other side. (Please refer to the 8X320 data sheet for complete details). Three external gates and the 8X320 allows the 8X401 to tie directly onto the 68010 bus, allowing host-display processor communications. Data and flag registers within the 8X320 allow this communication to be of command, data and semaphore type. As the 150-ns speed of the 8X401 exceeds the performance specifications of 8X300 family peripherals, the 8X401 must be slowed down whenever an access upon 8X300 peripherals is performed. The 8X401 Slow Clock Request (SCR) input is used to cause half speed 8X401 instruction execution. This input is driven through a fast select bit, which must be asserted when accessing the 8X320.

The 8X401 has 13 general purpose registers. For additional temporary storage, a Signetics 8X353 32-byte RAM is used. To allow the interrupt handler to save the processor state, a Signetics 8X355 STACK is used. In order to minimize the time the update process is required to wait, the display process to complete the 8X355 is fast selected. SCR must also be asserted when accessing either the 8X353 or the 8X355.

An 8X401-Based Raster Graphics System

AN4011

HARDWARE SUMMARY

This complete 8X401 graphics system design requires 13 8X470s, and one each of 8X320, 8X353, and 8X355. The various peripherals are mapped at the addresses shown in Table 1. The total 8X401 graphics software is less than 4K words. Total instruction word size is 31 bits; 20 for the 8X401 and 11 for fast select bits (A definition of the fast select bits is given in Table 2). Four Signetics 82S321 4K × 8 PROMs implement the instruction word. These four PROMs provide 32 instruction bits. One extra instruction bit is available for future expansion.

Table 2. System Fast Select Bits

b11 = unassigned, available for expansion
b10 = 8X355 STACK
b9 = TR*/QE*
b8 = B bank ADDR High
b7 = B bank ADDR Low
b6 = A bank ADDR High
b5 = A bank ADDR Low
b4 = B bank WRITE
b3 = B bank READ
b2 = A bank WRITE
b1 = A bank READ
b0 = Slow Clock Request

Table 1. Peripheral Addresses

Address (Hex)	Bank	Device
00	C	8X470 data register
30 – 3F	C	8X320
C0 – DF	C	8X353
F8	A & B	DATA0 8X470 data register
FA	A & B	DATA1 8X470 data register
FC	A & B	DATA2 8X470 data register
FE	A & B	DATA3 8X470 data register

SYSTEM OPERATION

System operation is straightforward. The main task of the 8X401 as a display processor is to update the picture contained in the display memory. Picture update commands are passed to the 8X401 by the host through the 8X320 mapped onto the C-bank. In order to update a picture, the 8X401 must specify a DRAM address (contained in ADDR HI, ADDR LOW) and manipulate DRAM data (contained in the data 8X470s). If an address 8X470 is accessed, its fast select bit must be asserted. If a data 8X470 is accessed, it must have been preselected through an 8X401 software select cycle. If either the 8X320, 8X353, or 8X355 are accessed, SCR must be asserted.

In order to define a READ* or WRITE* access on the display memory, the appropriate fast select bit must be asserted. Whenever the 8X401 is interrupted, a row access on the display DRAMs must be made. In order to define a row access on the display memory, the TR*/DE* and READ fast select bits must be asserted.

SOFTWARE

With a simple interface between the 8X401 and the display memory, attention can be turned to the graphics software. The 8X401 has a powerful bit-manipulation oriented instruction set (Table 3 lists the 32 basic instructions of the 8X401) that is ideal for the type of manipulations required in graphics software.

Table 3. 8X401 Instruction Set

INSTRUCTION TYPE	VARIATION	DESCRIPTION
ADD	ADD ADC AD8 AD5	ADD ADD with Carry ADD Immediate 8 ADD Immediate 5
AND	AND AN8 AN5	AND AND Immediate 8 AND Immediate 5
JUMP	JIF NS JIF S JIF NC JIF C JIF Z JIF NZ JSR JMP PSJ	JUMP IF SI = 0 JUMP IF SI = 1 JUMP IF Carry = 0 JUMP IF Carry = 1 JUMP IF ALU = 0 JUMP IF ALU<>0 JUMP to Subroutine JUMP POP STACK AND JUMP
MOVE	MOV	MOVE
RETURN	RIF NS RIF S RIF NC RIF C RIF Z RIF NZ RTN RCC RSC	RETURN IF SI = 0 RETURN IF SI = 1 RETURN IF Carry = 0 RETURN IF Carry = 1 RETURN IF ALU = 0 RETURN IF ALU<>0 RETURN RETURN and Clear Carry RETURN and Set Carry
XEC	XEC	Execute
XMIT	XT8 XT5	Transmit Immediate 8 Transmit Immediate 5
XOR	XOR XR8 XR5	Exclusive OR Exclusive OR Immediate 8 Exclusive OR Immediate 5

The 8X401, in addition to its powerful instruction set, has 13 general purpose registers and an on-board address stack which allows interrupt and subroutine capability (Figure 12 gives a block diagram of the 8X401 and Table 4 lists its registers.). Coupling the unique architecture of the 8X401 with its bit-manipulation oriented instruction set allows powerful graphics code to be written.

Table 4. Registers of the 8X401

REGISTER	ASSIGNMENT
0	Auxiliary Register
1 – A	General Purpose
B	XEC Displacement and General Purpose
C	Bank A Address Select
D	Bank B Address Select
E	Bank C Address Select
F	AND, XOR Immediate Destination, General Purpose
10	Carry Register
11	Data Access Banks A & B
12	Data Access Banks A & C
13	Data Access Bank A
14	Data Access Banks B & C
15	Data Access Bank B
16	Data Access Bank C
17	Status Register
18 – 1F	Left Rotate Override

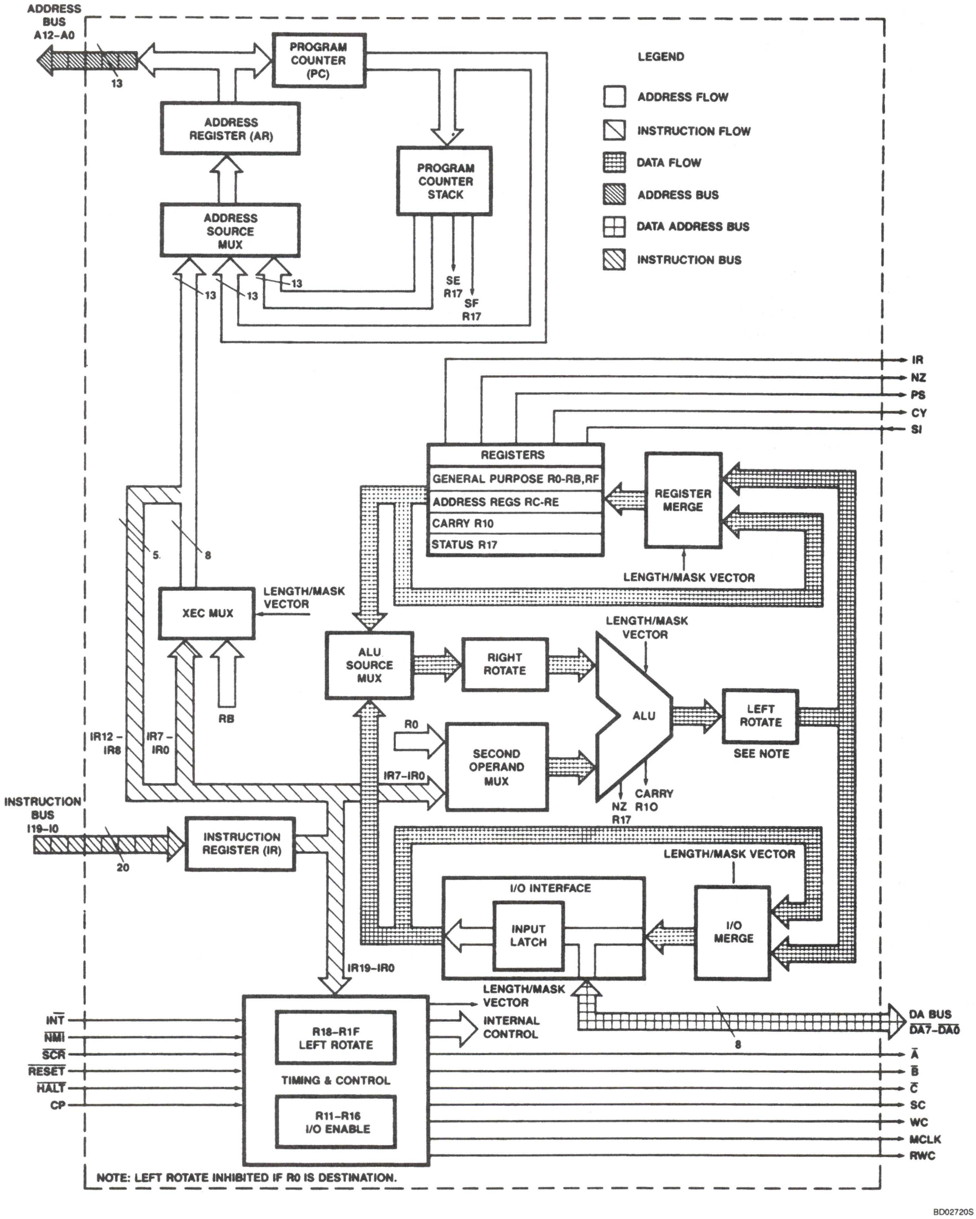

ADDRESS BUS A12–A0
13
PROGRAM COUNTER (PC)
ADDRESS REGISTER (AR)
PROGRAM COUNTER STACK
ADDRESS SOURCE MUX
13
13
13
SE R17
SF R17
LEGEND
ADDRESS FLOW
INSTRUCTION FLOW
DATA FLOW
ADDRESS BUS
DATA ADDRESS BUS
INSTRUCTION BUS
IR
NZ
PS
CY
SI
REGISTERS
GENERAL PURPOSE R0-RB,RF
ADDRESS REGS RC-RE
CARRY R10
STATUS R17
REGISTER MERGE
LENGTH/MASK VECTOR
5
8
XEC MUX
LENGTH/MASK VECTOR
RB
ALU SOURCE MUX
RIGHT ROTATE
LENGTH/MASK VECTOR
ALU
LEFT ROTATE
SEE NOTE
R0
SECOND OPERAND MUX
IR12 – IR8
IR7 – IR0
IR7–IR0
CARRY
NZ R10
R17
INSTRUCTION BUS I19–I0
20
INSTRUCTION REGISTER (IR)
LENGTH/MASK VECTOR
I/O INTERFACE
INPUT LATCH
I/O MERGE
IR19–IR0
LENGTH/MASK VECTOR
INTERNAL CONTROL
DA BUS DA7–DA0
8
INT
NMI
SCR
RESET
HALT
CP
R18–R1F LEFT ROTATE
TIMING & CONTROL
R11–R16 I/O ENABLE
A
B
C
SC
WC
MCLK
RWC
NOTE: LEFT ROTATE INHIBITED IF R0 IS DESTINATION.
BD02720S

Figure 12. Block Diagram of the 8X401

An 8X401-Based Raster Graphics System

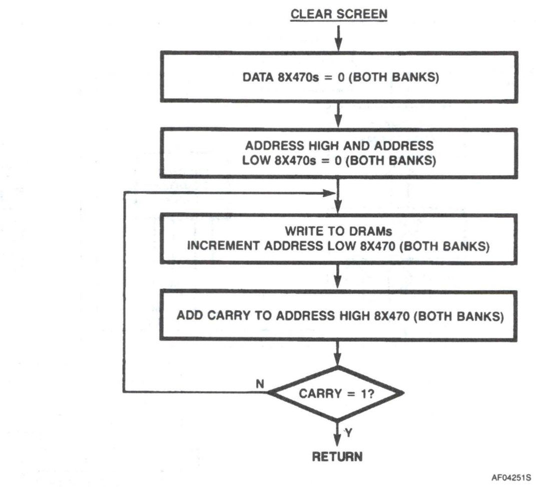
AN4011

For example, one host command the 8X401 is required to execute is PAINT. A specified portion of the screen is to be painted a specific color. A subset of the PAINT routine is the CLEAR SCREEN routine. In clearing the screen, 0000 is written to every pixel in the display memory.

Since the display memory is segmented into two parts, it would be ideal to clear both banks simultaneously. Without the ability to write to both halves at once, the entire memory would have to be cleared sequentially, doubling the time required to complete the routine. Writing to both banks requires the addresses in the address registers on both banks to be updated. With the 8X401's read-modify-write capability, coupled with its dual bank output capabilities, the addresses in the address registers on both banks can be updated in only two instructions.

The first of the two update instructions will read an address from ADDR Low on the A bank. The 8X401 will increment ADDR Low by one and output the result back to ADDR Low on both the A and B banks. The second instruction will read an address from ADDR High on the A bank. It will add the carry from the ADDR Low operation to it and output the result to ADDR High on both the A and B banks.

In addition to the above two instructions, an instruction is required for bounds checking. The algorithm for the CLEAR SCREEN routine is given in Flowchart 1. Table 5 gives the actual 8X401 code required to implement Flowchart 1. The inner loop of the clear screen routine is executed once for every 16 pixels. With 1M pixels in the system, the loop is executed $(1024)^2/16 = 65536$ times. With 3 instructions per loop, a total of 196,608 instructions are executed. The instructions outside the loop are insignificant when compared to the total loop instructions executed, making the total execution time for the clear screen routine about 29.6ms (19.7K instructions * 150ns/instruction = 29.6ms. The average pixel clearing rate is then about: $(1024)^2$pixels/29.6ms = 35.4M pixels/second. Note that this rate (and all succeeding rates) are for 4-bits per pixel.


CLEAR SCREEN
DATA 8X470s = 0 (BOTH BANKS)
ADDRESS HIGH AND ADDRESS LOW 8X470s = 0 (BOTH BANKS)
WRITE TO DRAMs
INCREMENT ADDRESS LOW 8X470 (BOTH BANKS)
ADD CARRY TO ADDRESS HIGH 8X470 (BOTH BANKS)
N
CARRY = 1?
Y
RETURN
AF04251S

Flowchart 1. Clear Screen Algorithm

Table 5. Clear Screen Code

```
CLEAR_SCREEN:                  XT8   #0, R0       *R0 = 0
                               XT8   #F8, RC      *select DATA0 8X470 on A bank
                               XT8   #F8, RD      *select DATA0 8X470 on B bank
                               XT8   #0, R11      *above 8X470s = 0
                               XT8   #FA, RC      *
                               XT8   #FA, RD      *as above except
                               XT8   #0, R11      *  DATA1 8X470
                               XT8   #FC, RC      *
                               XT8   #FC, RD      *as above except
                               XT8   #0, R11      *  DATA2 8X470
                               XT8   #FE, RC      *
                               XT8   #FE, RD      *as above except
                               XT8   #0, R11      *  DATA3 8X470
                               XT8   #01, RC      *deselect A bank
                               XT8   #01, RD      *deselect B bank
                  FS ABAR_L = 1  XT8   #0, R11      *ADDR Low on A, B banks = 0
                  FS ABAR_H = 1  XT8   #0, R11      *ADDR High on A, B banks = 0
(* Falling edge of WRITE* latches address at DRAM controller *)
LOOP:             FS WRITE = 1   ADI   R11, 1       *incr ADDR Low on A, B banks
                  FS ABAR_L = 1
                  FS ABAR_H = 1  ADC   R11, R11     *add carry to ADDR High on
                                                     A, B banks
                               RIF   C, LOOP      *if entire screen not cleared go
                                                     back
                                                  *done
EXIT:
```

Table 6. Clear Screen Code Without Fast Select

```
LOOP:   FS WRITE = 1   ADI   R11, 1              *incr ADDR Low on A, B banks
                       XT8   #ADDR High, RC      *select ADDR High on A bank
                       XT8   #ADDR High, RD      *select ADDR High on B bank
                       ADC   R11, R11            *add carry to ADDR High on
                                                  A, B banks
                       XT8   #ADDR Low, RC       *select ADDR Low on A bank
                       XT8   #ADDR Low, RD       *select ADDR Low on B bank
                       RIF   C, LOOP             *if entire screen not cleared go
                                                  back
EXIT:                                            *done
```

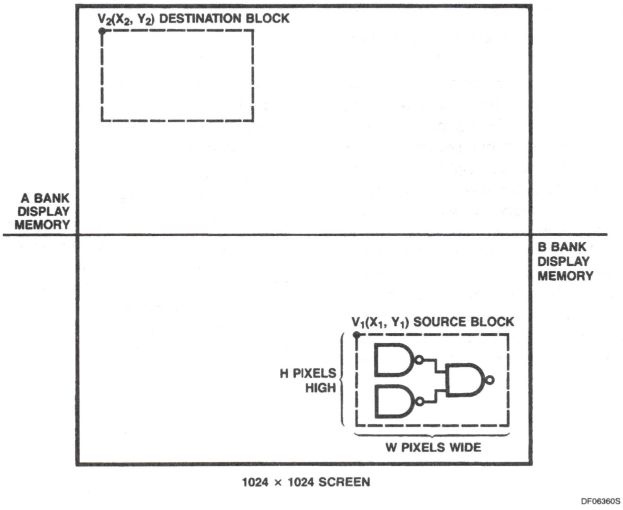
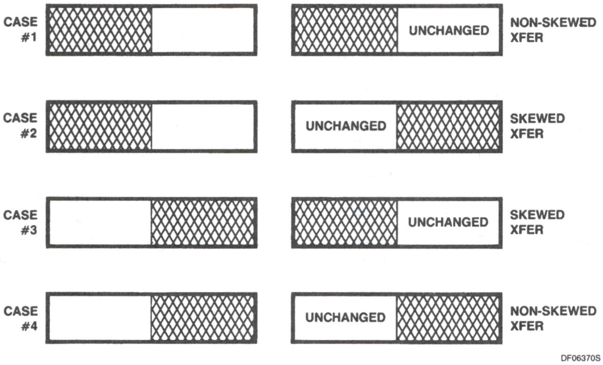


Figure 13. Bit-Block Transfer (BITBLT) Example

The use of fast select to select the address registers allows the update of the address registers to overlap the memory access time. Without fast select, four additional instructions would be required; two for ADDR Low selection (one for each bank), and two for ADDR High (one for each bank). The inner loop of the clear screen code for non-fast select would be that shown in Table 6. The average pixel clearing rate in the non-microcode case would be about 15.2M pixels/second. Thus, 57% speed improvement has been realized using fast selects of the address registers.

Another host instruction the system is required to execute is bit-block transfer (BIT – BLT). The 8X401 will transfer a block w pixels wide and h pixels high at vertex v1 (with logical address < x1,y1 >) to vertex v2 (with logical address < x2,y2 >). The BIT – BLT command; the vertex addresses v1 and v2; and the parameters w and h are all transferred through the host 8X470s on the C bank. An example of BIT – BLT is shown in Figure 13. The block on the the bottom part of the screen is to be transferred to the block on the top part of the screen. Since the display memory was divided into two parts, the top half of the screen resides in the A bank display memory, while the bottom half of the screen is in the B bank display memory.

The source and destination pixels in the BIT – BLT of Figure 13 are in packed format, with two pixels per data 8X470. Depending on how the source and destination pixels are aligned, the transfer is one of the four cases shown in Figure 14.



NOTE:
Cross-hatched area in the square 8X470 is the pixel to be moved. Cross-hatched area in the destination 8X470 is where the square pixel is to be moved.

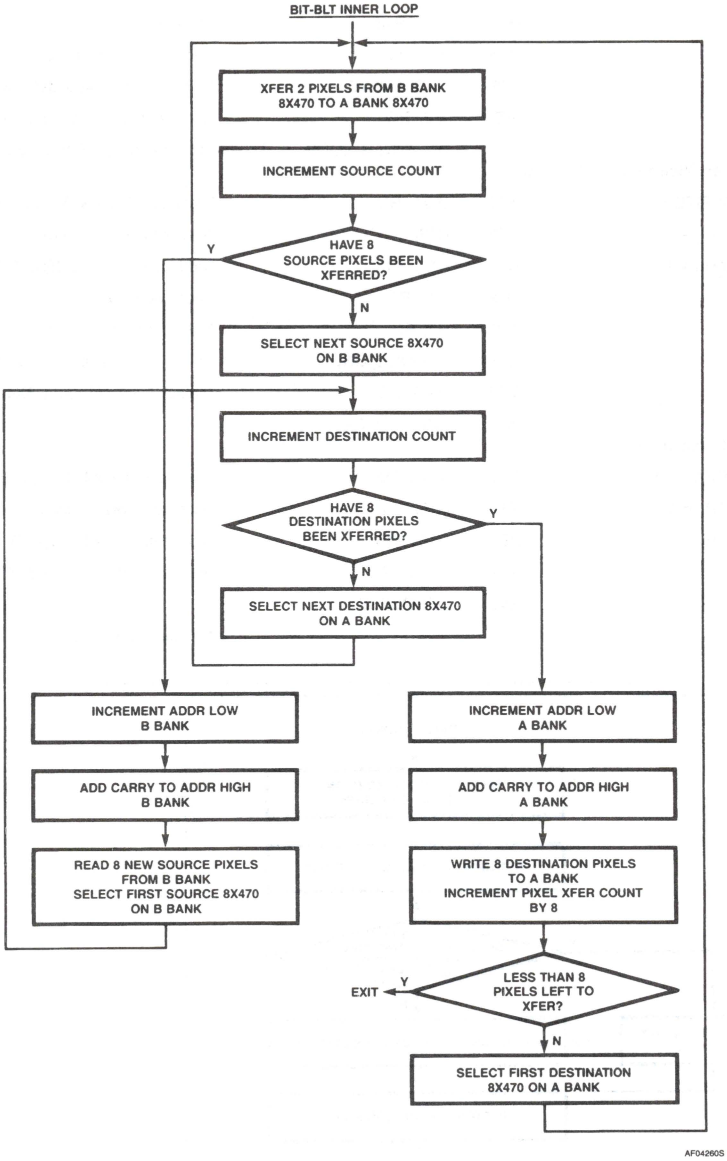
Figure 14. The Four Pixel Transfer Possibilities

The example in Figure 13 shows the source and destination pixels aligned along byte boundaries (case one of four). The algorithm for the inner loop of this bank-to-bank byte boundary BIT – BLT is given in Flowchart 2. This routine will transfer the source block along raster lines. A pseudo-code representation of this transfer is:

```
FOR I: = 1 to h lines DO
    transfer w pixels from x1 along line
    (y1 + I);
```

Table 7 gives the code for the inner loop of the BIT – BLT routine. The loop which will transfer pixels along raster lines selects a data 8X470 on both the B bank and A bank, and then transfers a byte of data from bank B to bank A. The inner loop requires one READ of B bank DRAMs per eight pixels and one WRITE to A bank DRAMs per eight pixels. As before, the memory access times can be overlapped with other functions. The inner loop in Table 7 requires 33 instructions per eight pixels. When more than a few pixels are transferred, the loop is executed a multiple of times. As was the case with the clear screen routine, the instructions outside the loop are insignificant compared to the total. The total average pixel transfer time is about: 4950-ns/ 8 pixels = 619-ns/pixel. The average pixel transfer rate is: 1/ < 619-ns/pixel > = 1.62M pixels/second.

In addition to the host commands required to be executed by the 8X401, the 8X401 must also refresh the CRT screen. As mentioned previously, the 8X401 is interrupted once every 15.4-μs to generate a row access on the display memory. Upon receiving an interrupt, the 8X401 will set the interrupt mask, push the program counter onto the stack and initiate the interrupt handler. This procedure takes 300ns.

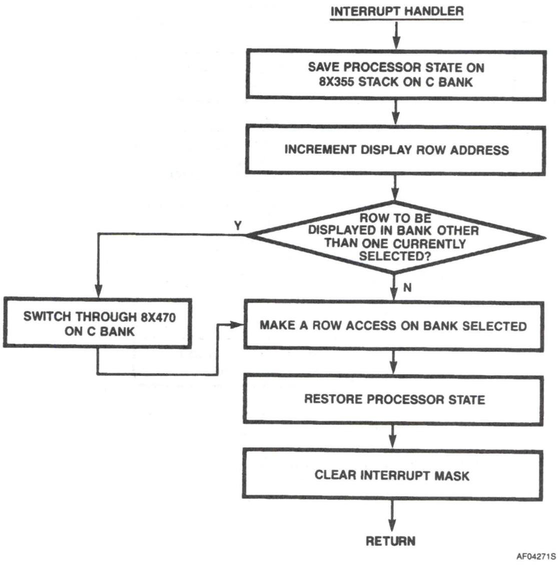

BIT-BLT INNER LOOP
XFER 2 PIXELS FROM B BANK 8X470 TO A BANK 8X470
INCREMENT SOURCE COUNT
HAVE 8 SOURCE PIXELS BEEN XFERRED?
Y
N
SELECT NEXT SOURCE 8X470 ON B BANK
INCREMENT DESTINATION COUNT
HAVE 8 DESTINATION PIXELS BEEN XFERRED?
Y
N
SELECT NEXT DESTINATION 8X470 ON A BANK
INCREMENT ADDR LOW B BANK
ADD CARRY TO ADDR HIGH B BANK
READ 8 NEW SOURCE PIXELS FROM B BANK SELECT FIRST SOURCE 8X470 ON B BANK
INCREMENT ADDR LOW A BANK
ADD CARRY TO ADDR HIGH A BANK
WRITE 8 DESTINATION PIXELS TO A BANK INCREMENT PIXEL XFER COUNT BY 8
LESS THAN 8 PIXELS LEFT TO XFER?
Y
EXIT
N
SELECT FIRST DESTINATION 8X470 ON A BANK
AF04260S

Flowchart 2. Inner Loop of BITBLT of Figure 6

Table 7. BIT – BLT Inner Loop Code

```
INNER_LOOP:                       MOV   R15, R13          *mov B bank to A bank
                                  ADI   #2, RD            *incr read count by 2 & select B bank 8X470
                                  JIF   C, SUPDATE        *if no more source data get some more
DPROCESS:                         ADI   #2, RC            *incr write count by 2 & select A bank 8X470
                                  JIF   NC, INNER_LOOP    *if 8 pixels not Xferred then Xfer more
(* 8 pixels have been Xferred so write them *)
DUPDATE:   FS AWRITE = 1          XTB   #01, RC           *deselect A bank & write A bank
           FS AAR_L = 1           ADI   R13, 1            *incr addr low A bank
           FS AAH_H = 1           ADC   R13, R13          *add carry to addr high A bank
                                  ADI   #8, R1            *incr count low
                                  ADC   R2, R2            *add carry count high
                                  JIF   C, DONE_LINE      *if < 8 pixels left to Xfer then exit loop
                                  XT8   #F8, RC           *select least sig. data 8X470 A bank
                                  JMP   INNER_LOOP
(* Read 8 pixels of source data *)
SUPDATE:   FS BREAD = 1           XT8   #01, RD           *deselect B bank & read B bank
           FS BAR_L = 1           ADI   R15, 1            *incr addr low B bank
           FS BAR_H = 1           ADC   R15, R15          *add carry to addr high B bank
                                  XT8   #F8, RD           *select least sig. data 8X470 B bank
                                  JMP   DPROCESS

                                                          *process next line
DONE_LINE:
```


INTERRUPT HANDLER
SAVE PROCESSOR STATE ON 8X355 STACK ON C BANK
INCREMENT DISPLAY ROW ADDRESS
ROW TO BE DISPLAYED IN BANK OTHER THAN ONE CURRENTLY SELECTED?
Y
N
SWITCH THROUGH 8X470 ON C BANK
MAKE A ROW ACCESS ON BANK SELECTED
RESTORE PROCESSOR STATE
CLEAR INTERRUPT MASK
RETURN
AF04271S

Flowchart 3. Interrupt Handler Algorithm

The interrupt handler will save portions of the interrupted state upon entry. It will then make a row access upon the display memory, switching display banks (through the 8X470 on bank C) if required. It will finally restore the interrupted state and clear the interrupt mask before exiting. The algorithm for the interrupt handler is given in Flowchart 3.

The total time spent in servicing an interrupt adds up to about 4.05μs. The display overhead is then: 4.05μs/15.4μs = 26%. The update process is kept from accessing the display memory for picture update only 26% of the time. Compared to the 75% wait time associated with other designs, this is a significant improvement.

It should be noted that the interrupt handler does not have to save the data contained in the data 8X470s on the active display bank. This is due to the fact that a row access does not affect the data contained in the data 8X470s. This operation can briefly be explained as follows: The 8X470's user port has two control signals, input control and output control (IC* and OC*). Upon the initiation of a DRAM write access, the hardware in the READ*/WRITE* logic assets OC*, writing the data contained within the data 8X470s to the DRAMs.

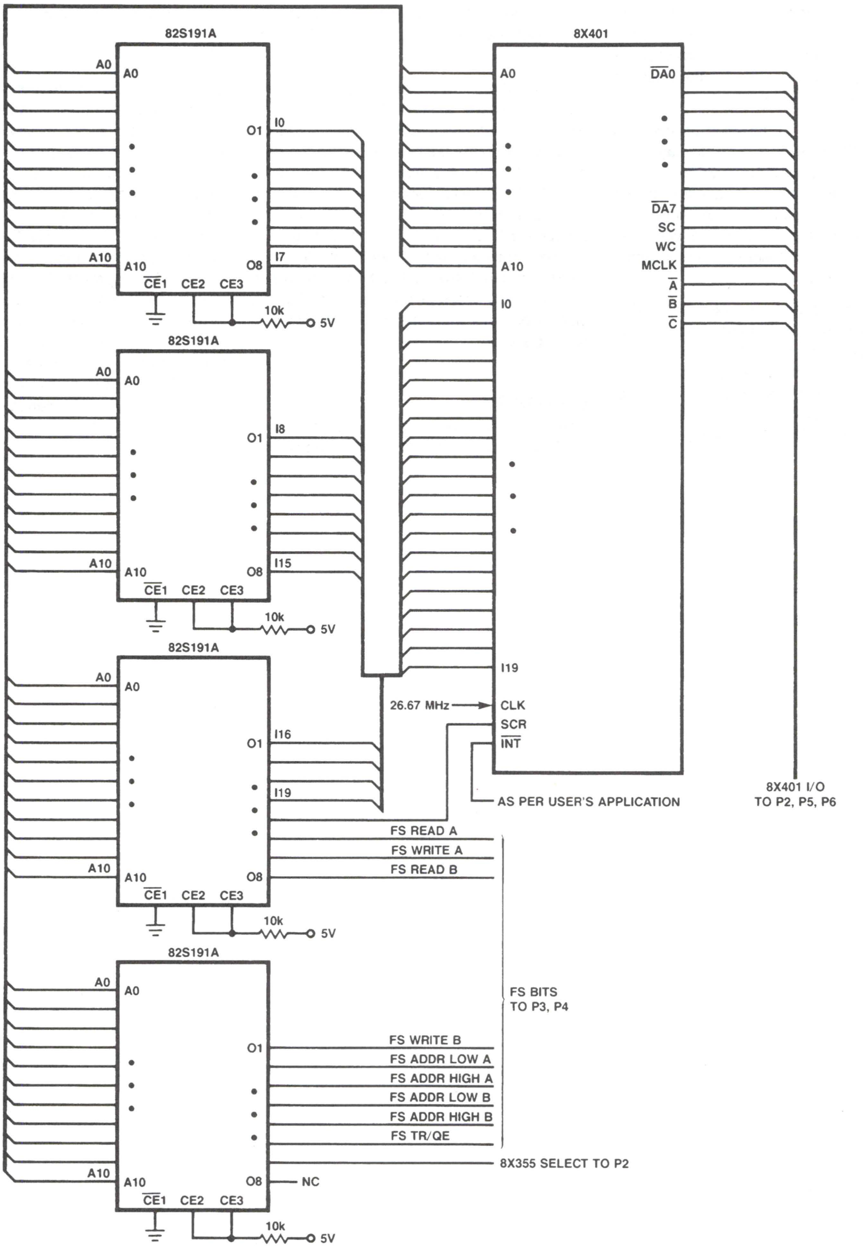
Upon initiation of a DRAM read access, this same hardware asserts IC*, opening the 8X470 data latches to DRAM data. When IC* is deasserted, the latches close and the data is latched. With a row access, both the READ and TR*/QE* fast select bits will be asserted. This combination of bits prevents the READ*/WRITE* logic from asserting IC*, thereby ensuring the integrity of the data within the data 8X470s. This hardware design simplifies the interrupt handler, reducing the display process wait time and improving system operation.

SUMMARY

The two basic performance limitations in traditional raster scan graphics systems typically have been the display processor and the display memory. New-generation video DRAMs serve to reduce the memory bottlenecks present in raster graphics systems.

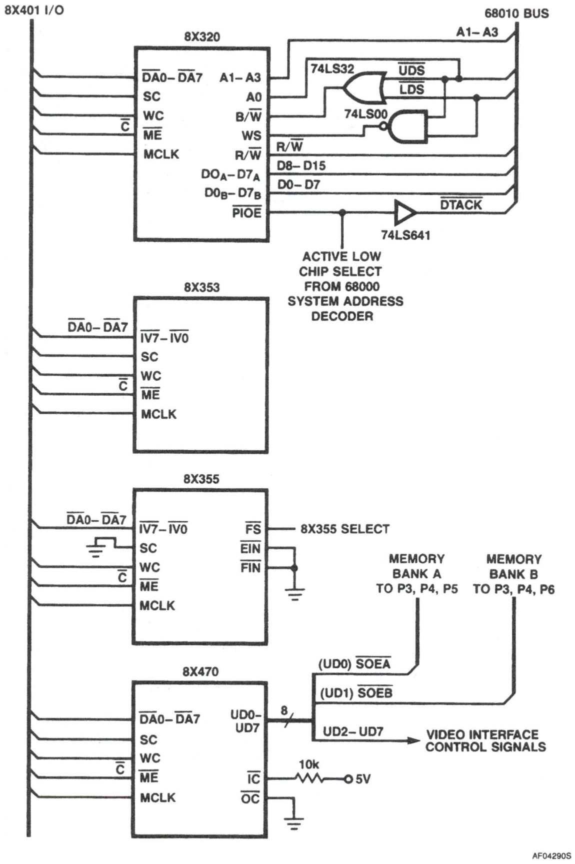
The 8X401, with its 150ns speed, its 13 general-purpose registers, and its 32 basic bit-manipulation instructions, reduces the second performance limitation, namely that of the display processor. With complete peripheral support available in the 8X400 family, as well as the 8X300 family, system hardware design is simplified and chip counts are reduced. Software support in the form of the 8X401 FORTRESS cross-assembler simplifies software development efforts. Finally, the 8X401 Development Board completes the development phase by allowing complete hardware and software system integration. This triad of hardware, software and development support easily allows the 8X401 microcontroller to serve as the heart of any high-resolution, high-performance raster graphics system.

APPLICATION BOARD SCHEMATIC


82S191A
A0
A10
O1
O8
CE1 CE2 CE3
10k
5V
I0
I7
I8
I15
I16
I19
8X401
A0
A10
I0
I19
DA0
DA7
SC
WC
MCLK
A
B
C
26.67 MHz
CLK
SCR
INT
AS PER USER'S APPLICATION
8X401 I/O
TO P2, P5, P6
FS READ A
FS WRITE A
FS READ B
FS WRITE B
FS ADDR LOW A
FS ADDR HIGH A
FS ADDR LOW B
FS ADDR HIGH B
FS TR/QE
FS BITS
TO P3, P4
8X355 SELECT TO P2
NC
AF04280S

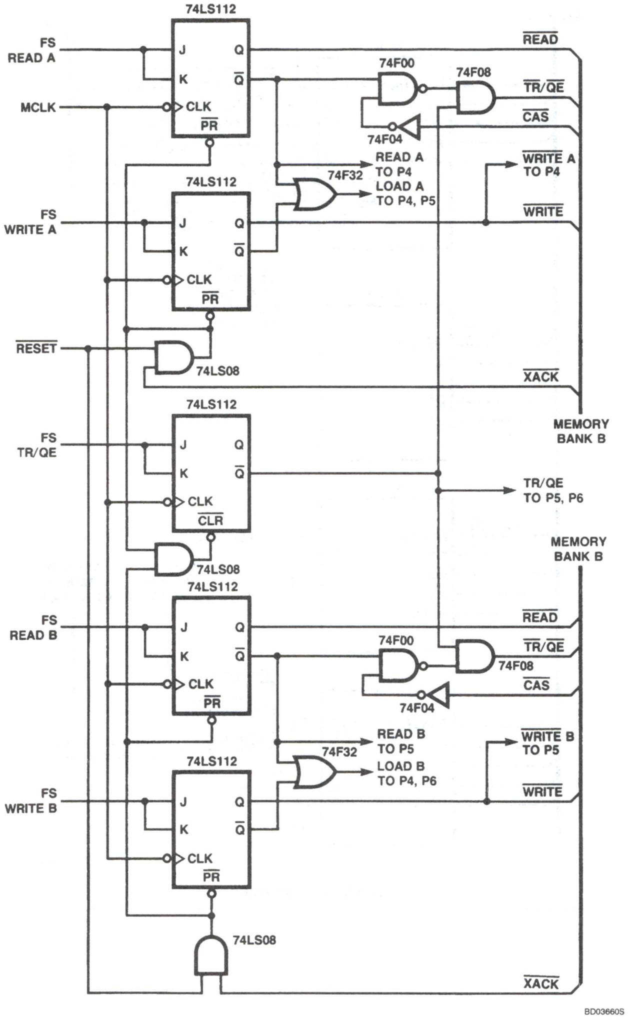
8X401 Subsystem

APPLICATION BOARD SCHEMATIC (Continued)


8X401 I/O
68010 BUS
A1–A3
8X320
DA0–DA7
SC
WC
C
ME
MCLK
A1–A3
A0
B/W
WS
R/W
DOA–D7A
D0B–D7B
PIOE
74LS32
UDS
LDS
74LS00
R/W
D8–D15
D0–D7
DTACK
74LS641
ACTIVE LOW CHIP SELECT FROM 68000 SYSTEM ADDRESS DECODER
8X353
DA0–DA7
IV7–IV0
SC
WC
C
ME
MCLK
8X355
DA0–DA7
IV7–IV0
SC
WC
C
ME
MCLK
FS
EIN
FIN
8X355 SELECT
MEMORY BANK A TO P3, P4, P5
MEMORY BANK B TO P3, P4, P6
8X470
(UD0) SOEA
(UD1) SOEB
DA0–DA7
SC
WC
C
ME
MCLK
UD0–UD7
8
UD2–UD7
VIDEO INTERFACE CONTROL SIGNALS
10k
5V
IC
OC
AF04290S

System Resources

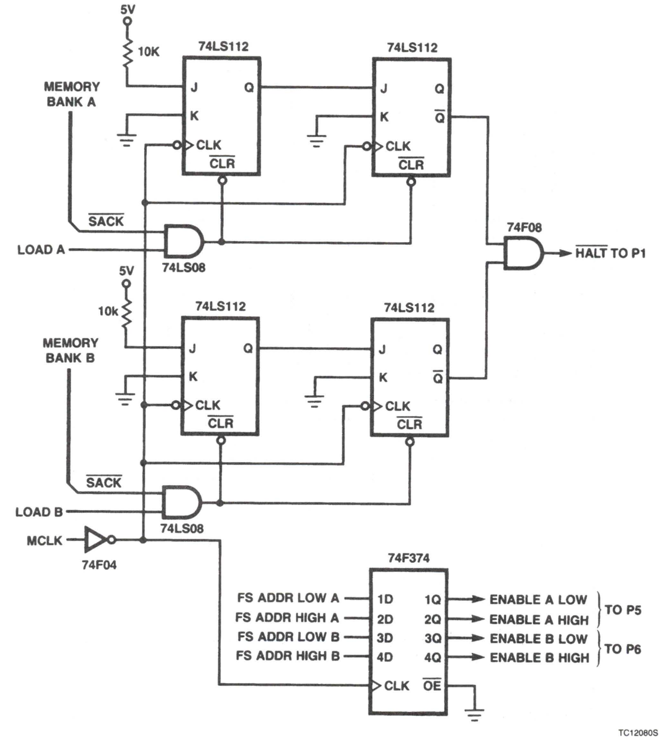
APPLICATION BOARD SCHEMATIC (Continued)


74LS112
FS READ A
J
K
Q
$\overline{Q}$
MCLK
CLK
$\overline{PR}$
74F00
74F08
74F04
$\overline{READ}$
$\overline{TR/QE}$
$\overline{CAS}$
READ A TO P4
74F32
LOAD A TO P4, P5
$\overline{WRITE A}$ TO P4
$\overline{WRITE}$
74LS112
FS WRITE A
$\overline{RESET}$
74LS08
$\overline{XACK}$
74LS112
FS TR/QE
$\overline{CLR}$
MEMORY BANK B
TR/QE TO P5, P6
MEMORY BANK B
74LS08
74LS112
FS READ B
$\overline{READ}$
74F00
$\overline{TR/QE}$
74F08
$\overline{CAS}$
74F04
READ B TO P5
74F32
LOAD B TO P4, P6
$\overline{WRITE B}$ TO P5
$\overline{WRITE}$
74LS112
FS WRITE B
74LS08
$\overline{XACK}$
BD03660S

Memory Control

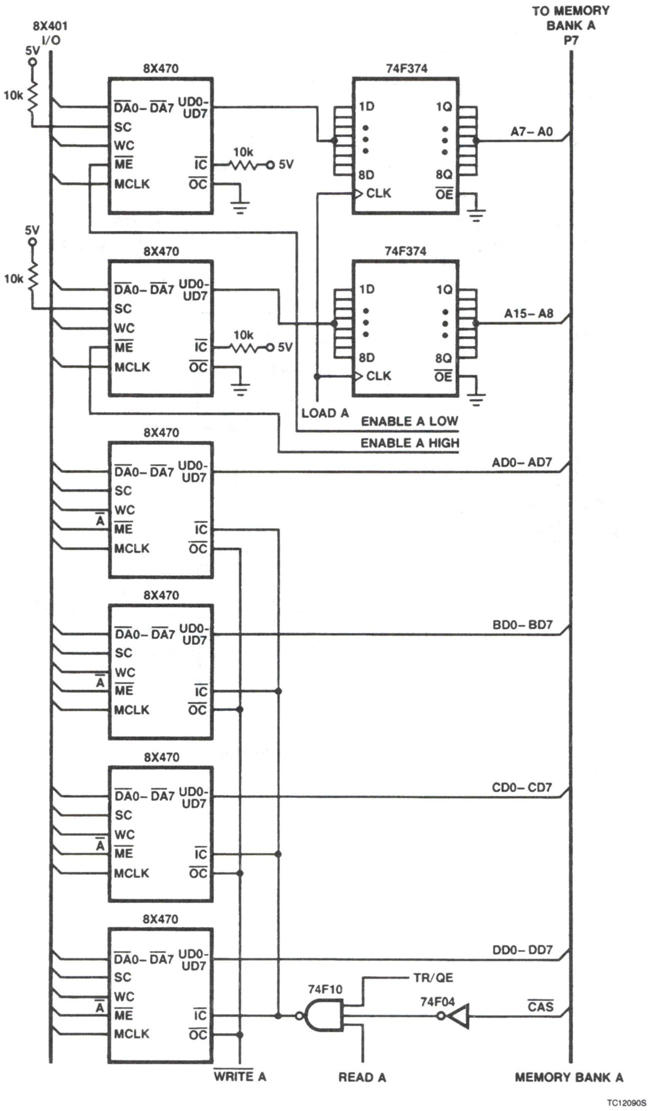
An 8X401-Based Raster Graphics System AN4011

APPLICATION BOARD SCHEMATIC (Continued)


5V
10K
74LS112
74LS112
MEMORY BANK A
J
Q
K
Q̄
CLK
CLR
SACK
LOAD A
74LS08
74F08
HALT TO P1
5V
10k
MEMORY BANK B
SACK
LOAD B
MCLK
74F04
74F374
FS ADDR LOW A
FS ADDR HIGH A
FS ADDR LOW B
FS ADDR HIGH B
1D
2D
3D
4D
1Q
2Q
3Q
4Q
OE
ENABLE A LOW
ENABLE A HIGH
ENABLE B LOW
ENABLE B HIGH
TO P5
TO P6
TC12080S

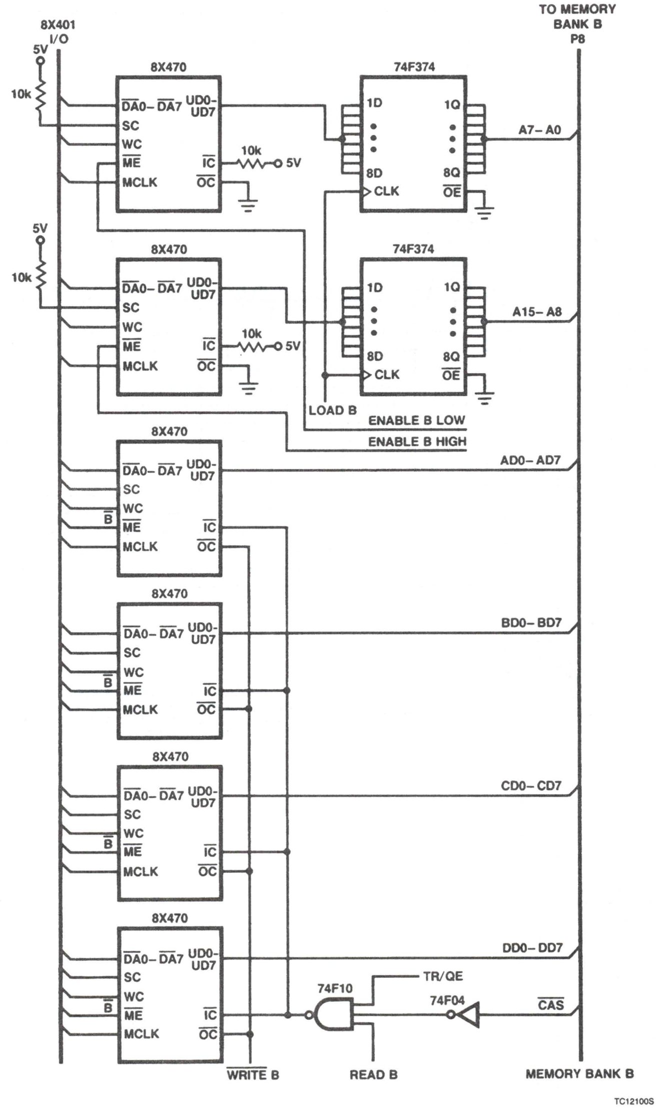
Halt and Memory Enables

APPLICATION BOARD SCHEMATIC (Continued)


TO MEMORY
BANK A
P7
8X401
I/O
5V
10k
8X470
$\overline{DA0}$–$\overline{DA7}$
UD0-
UD7
SC
WC
$\overline{ME}$
MCLK
$\overline{IC}$
$\overline{OC}$
10k
5V
74F374
1D
8D
CLK
1Q
8Q
$\overline{OE}$
A7– A0
A15– A8
LOAD A
ENABLE A LOW
ENABLE A HIGH
AD0– AD7
BD0– BD7
CD0– CD7
DD0– DD7
$\overline{A}$
74F10
TR/QE
74F04
$\overline{CAS}$
$\overline{WRITE}$ A
READ A
MEMORY BANK A
TC12090S

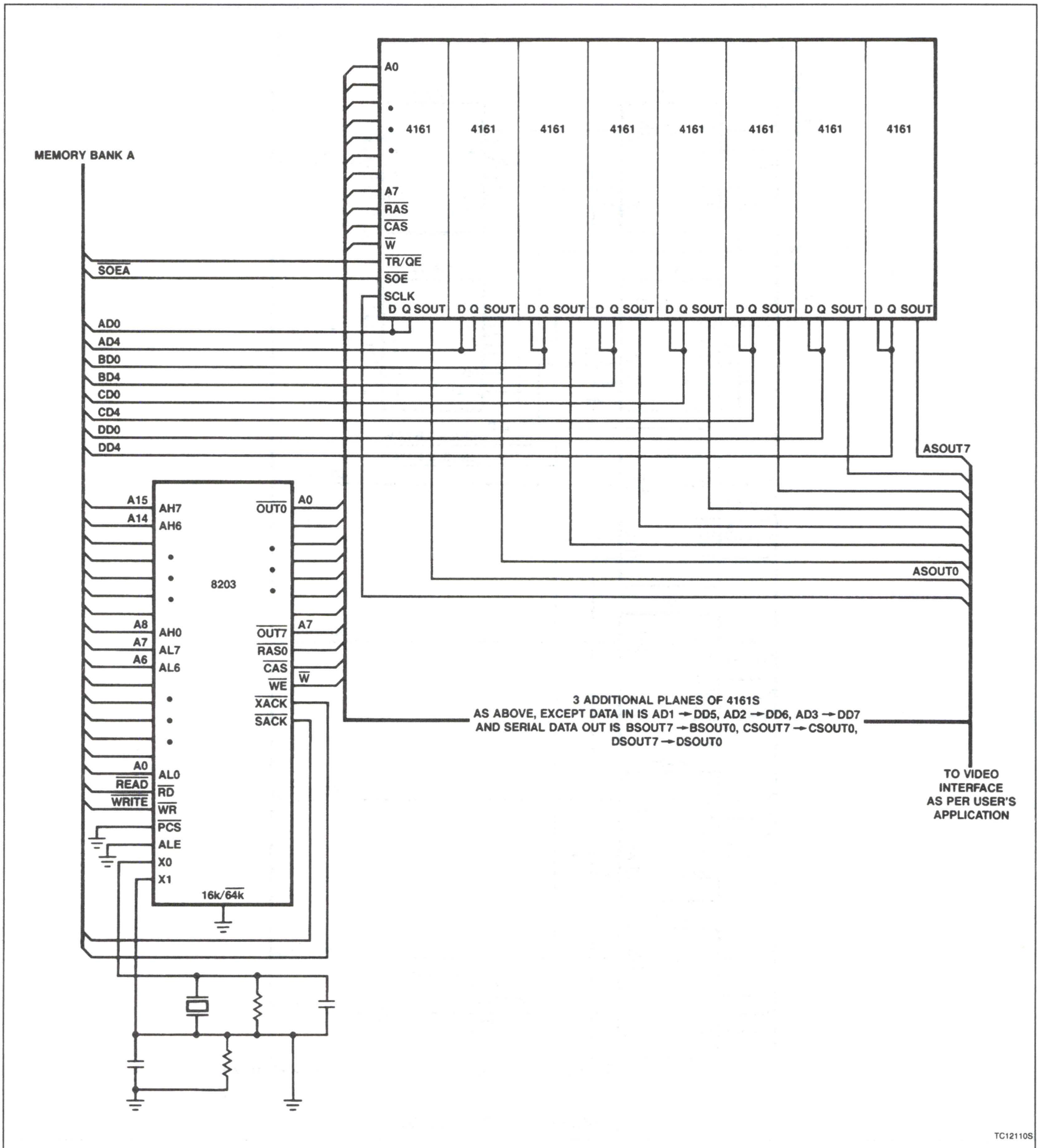
Memory Address and Data for Bank A

APPLICATION BOARD SCHEMATIC (Continued)


8X401
I/O
5V
10k
8X470
DA0– DA7
UD0–UD7
SC
WC
ME
MCLK
IC
OC
10k
5V
74F374
1D
1Q
8D
8Q
CLK
OE
TO MEMORY
BANK B
P8
A7– A0
A15– A8
LOAD B
ENABLE B LOW
ENABLE B HIGH
AD0– AD7
BD0– BD7
CD0– CD7
DD0– DD7
B
74F10
74F04
TR/QE
CAS
WRITE B
READ B
MEMORY BANK B
TC12100S

Memory Address and Data for Bank B

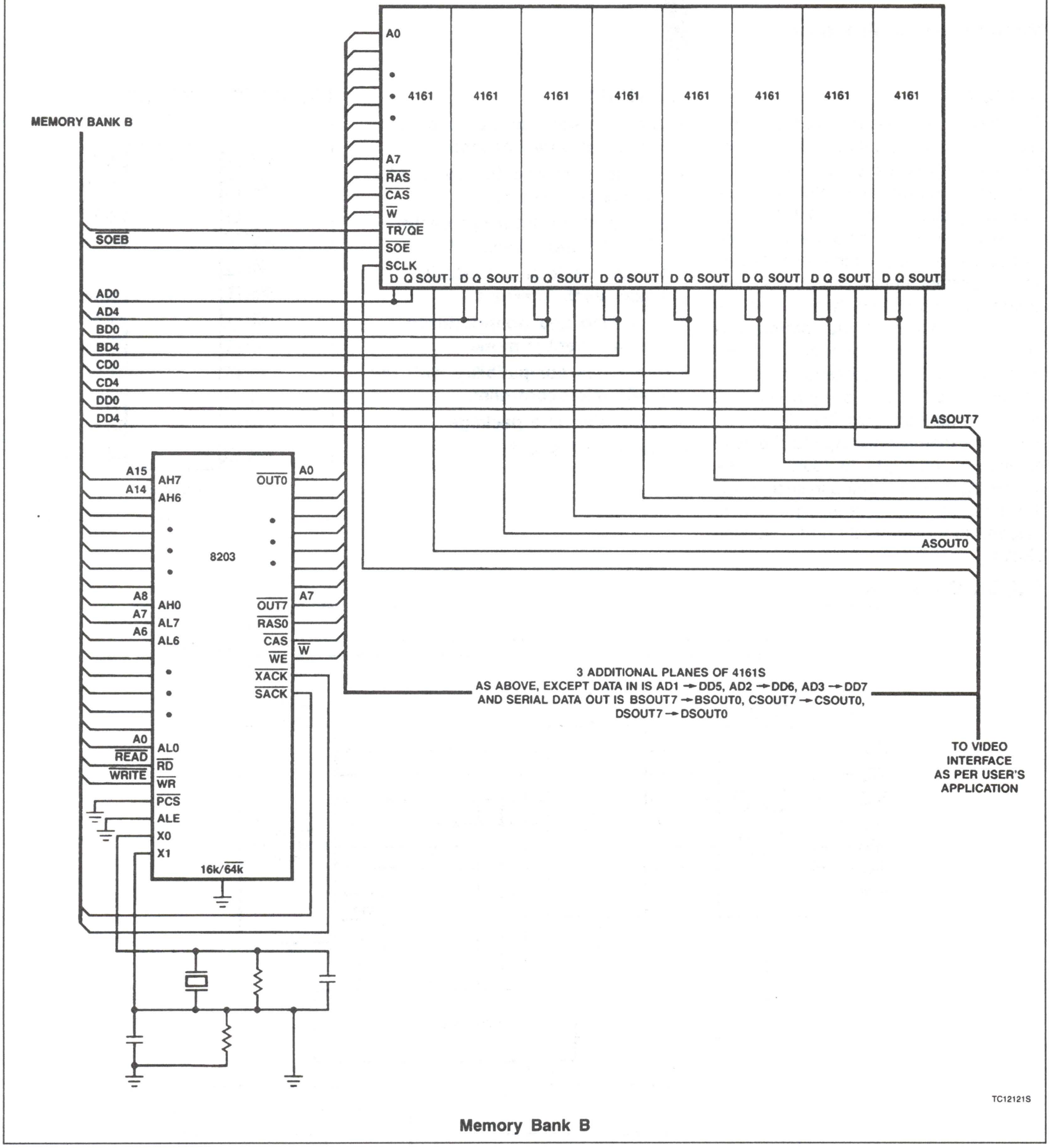
APPLICATION BOARD SCHEMATIC (Continued)


MEMORY BANK A
A0
A7
RAS
CAS
W
TR/QE
SOE
SCLK
4161
4161
4161
4161
4161
4161
4161
4161
D Q SOUT
SOEA
AD0
AD4
BD0
BD4
CD0
CD4
DD0
DD4
ASOUT7
ASOUT0
A15
A14
AH7
AH6
8203
OUT0
A0
A8
A7
A6
AH0
AL7
AL6
OUT7
RAS0
CAS
WE
XACK
SACK
A7
W
A0
READ
WRITE
AL0
RD
WR
PCS
ALE
X0
X1
16k/64k
TO VIDEO INTERFACE AS PER USER'S APPLICATION
TC12110S

3 ADDITIONAL PLANES OF 4161S
AS ABOVE, EXCEPT DATA IN IS AD1 → DD5, AD2 → DD6, AD3 → DD7
AND SERIAL DATA OUT IS BSOUT7 → BSOUT0, CSOUT7 → CSOUT0,
DSOUT7 → DSOUT0

Memory Bank A

APPLICATION BOARD SCHEMATIC (Continued)


MEMORY BANK B
4161
A0
A7
RAS
CAS
W
TR/QE
SOE
SCLK
D Q SOUT
SOEB
AD0
AD4
BD0
BD4
CD0
CD4
DD0
DD4
ASOUT7
ASOUT0
8203
A15 AH7
A14 AH6
A8 AH0
A7 AL7
A6 AL6
A0 AL0
READ RD
WRITE WR
PCS
ALE
X0
X1
16k/64k
OUT0 A0
OUT7 A7
RAS0
CAS
WE W
XACK
SACK
3 ADDITIONAL PLANES OF 4161S
AS ABOVE, EXCEPT DATA IN IS AD1 → DD5, AD2 → DD6, AD3 → DD7
AND SERIAL DATA OUT IS BSOUT7 → BSOUT0, CSOUT7 → CSOUT0,
DSOUT7 → DSOUT0
TO VIDEO INTERFACE AS PER USER'S APPLICATION
TC12121S

Memory Bank B

Signetics

8X411
Priority Interrupt Encoder

Objective Specification

Microprocessor Products

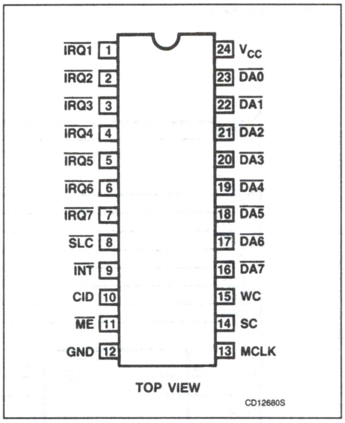
DESCRIPTION

The 8X411 Priority Interrupt Encoder provides a simple method to allow multiple interrupt sources to be connected to a single microprocessor interrupt input. Interfacing directly to the 8X401 microcontroller, the 8X411 generates an interrupt output in response to one of its own seven interrupt inputs, and provides the means by which the host processor determines which of the interrupts is pending. Two modes of operation allow interrupts to be processed either by predetermined priority or by a software controlled method. Two 8X411s can be cascaded together to expand the total number of interrupts to fourteen. The cascading is implemented in such a way that no additional time is expended to respond to the interrupts on the second 8X411.

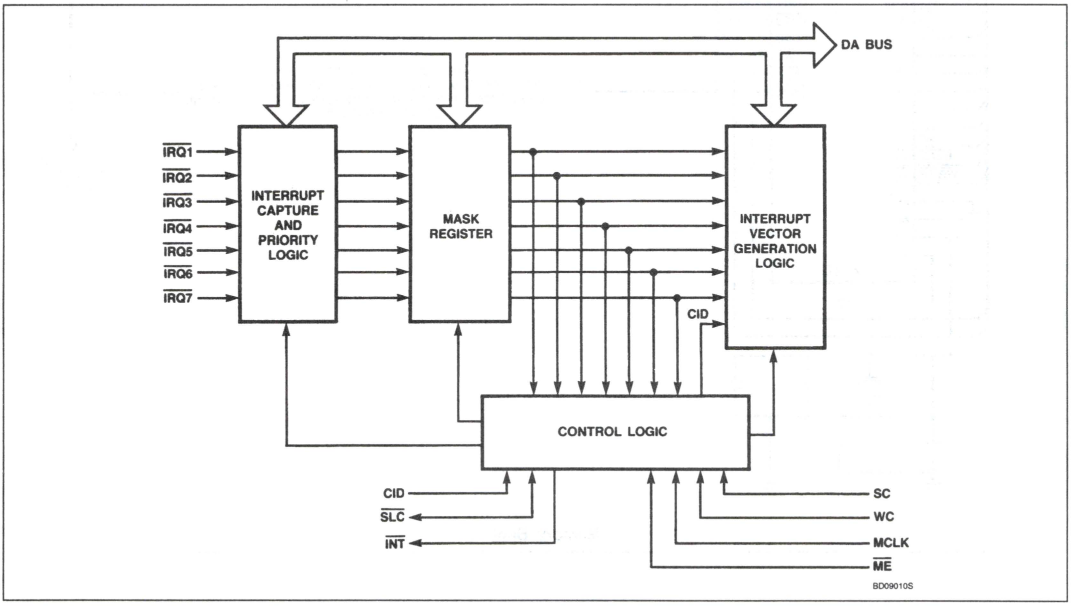
FEATURES

- **One chip controls up to seven prioritized interrupt inputs.**
- **Can be cascaded for up to 14 prioritized interrupts.**
- **Uses a single interrupt vector address when cascaded.**
- **A mask register individually masks each interrupt.**
- **Prioritized and nonprioritized interrupt vector modes.**
- **Direct bus compatibility with the 8X401 microcontroller.**
- **24-pin slim line package.**
- **Single +5V supply.**

PIN CONFIGURATION


IRQ1 1
IRQ2 2
IRQ3 3
IRQ4 4
IRQ5 5
IRQ6 6
IRQ7 7
SLC 8
INT 9
CID 10
ME 11
GND 12
24 V_{CC}
23 DA0
22 DA1
21 DA2
20 DA3
19 DA4
18 DA5
17 DA6
16 DA7
15 WC
14 SC
13 MCLK
TOP VIEW
CD12680S

BLOCK DIAGRAM


DA BUS
IRQ1
IRQ2
IRQ3
IRQ4
IRQ5
IRQ6
IRQ7
INTERRUPT CAPTURE AND PRIORITY LOGIC
MASK REGISTER
INTERRUPT VECTOR GENERATION LOGIC
CID
CONTROL LOGIC
CID
SLC
INT
SC
WC
MCLK
ME
BD09010S

Priority Interrupt Encoder 8X411

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin DIL plastic	N8X411N

PIN DESCRIPTION

PIN NO.	IDENTIFIER	FUNCTION
1 – 7	$\overline{IRQ1}$ – $\overline{IRQ7}$	**Interrupt Request:** User interrupt request lines. Active low.
8	$\overline{SLC}$	**Slave Control:** This pin is provides the handshaking between two cascaded 8X411s.
9	$\overline{INT}$	**Interrupt Output:** This active low, open collector output is the Interrupt output to the host processor.
10	CID	**Chip ID:** Input on this pin determines the 8X411 operation as either a Master or a Slave.
11	$\overline{ME}$	**Master Enable:** Controls operations on the DA bus, in conjunction with SC, WC, and MCLK.
12	GND	**Ground.**
13	MCLK	**Master Clock:** The system clock from the host microprocessor.
14	SC	**Select Command:** When high, indicates that an address is being output on the DA bus.
15	WC	**Write Command:** When high, indicates that data is being output on the DA bus.
16 – 23	$\overline{DA7}$ – $\overline{DA0}$	**DA Bus:** Bidirectional, 3-State, data and address bus.
24	V_{CC}	+5V power supply.

Signetics

8X450
RAM

Product Specification

Microprocessor Products

DESCRIPTION

The 8X450 is a 256-byte RAM designed primarily as a working storage element for 8X305 based systems. The 8X450 can be used as a dedicated RAM or in groups of RAMs to form a customized memory stack.

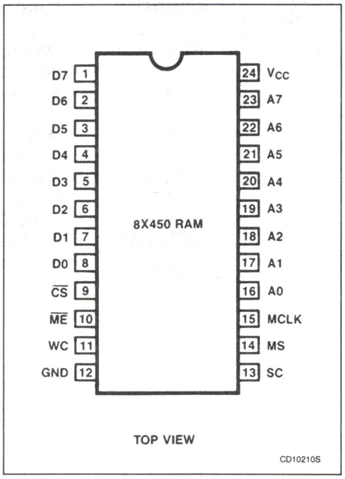
FEATURES

- **256-byte RAM**
- **Two modes of operation**
- **Paged RAM capability**
- **On-chip address latches**
- **Directly bus compatible with the 8X305 microcontroller**
- **24-pin slim line package**
- **Single +5V supply**

ORDERING INFORMATION

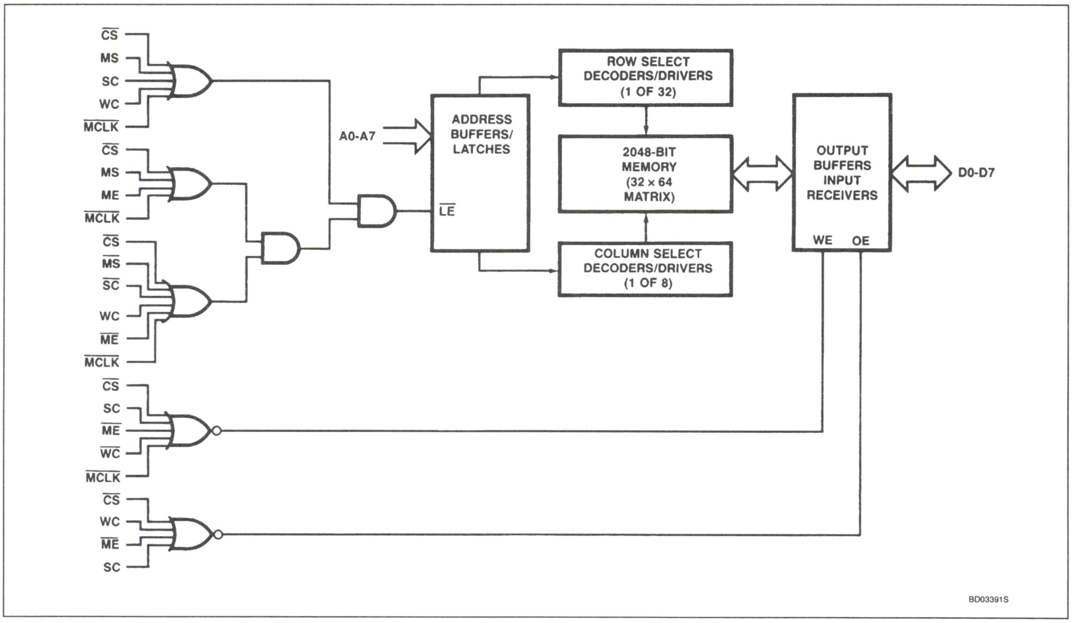
DESCRIPTION	ORDER CODE
24-Pin Ceramic DIP 400 Mil Wide	N8X450I

PIN CONFIGURATION



D7 1
D6 2
D5 3
D4 4
D3 5
D2 6
D1 7
D0 8
$\overline{CS}$ 9
$\overline{ME}$ 10
WC 11
GND 12
24 V_{CC}
23 A7
22 A6
21 A5
20 A4
19 A3
18 A2
17 A1
16 A0
15 MCLK
14 MS
13 SC
8X450 RAM
TOP VIEW
CD10210S

8X450 BLOCK DIAGRAM



$\overline{CS}$
MS
SC
WC
$\overline{MCLK}$
$\overline{CS}$
MS
ME
$\overline{MCLK}$
$\overline{CS}$
$\overline{MS}$
$\overline{SC}$
WC
$\overline{ME}$
$\overline{MCLK}$
$\overline{CS}$
SC
$\overline{ME}$
$\overline{WC}$
$\overline{MCLK}$
$\overline{CS}$
WC
$\overline{ME}$
SC
A0-A7
ADDRESS BUFFERS/ LATCHES
$\overline{LE}$
ROW SELECT DECODERS/DRIVERS (1 OF 32)
2048-BIT MEMORY (32 x 64 MATRIX)
COLUMN SELECT DECODERS/DRIVERS (1 OF 8)
OUTPUT BUFFERS INPUT RECEIVERS
WE
OE
D0-D7
BD03391S

RAM 8X450

FUNCTIONAL OPERATION

The 8X450 can be used in either of two operating modes — conventional DA select or fast select. Each mode supports either single or multiple RAM applications. The mode of operation is selected by the MS pin.

Conventional $\overline{DA}$ Select (MS = 1)

When MS is tied high, the conventional $\overline{DA}$ addressing mode is selected; in this mode, the 8X450 data and address lines are tied together. For applications that use a single RAM, the chip select ($\overline{CS}$) input is grounded. To prevent bus contention in applications that use multiple RAMs, the $\overline{CS}$ input is driven by an external source. Examples of both configurations (single and multiple RAMs) are shown in the Application Diagrams at the rear of this data sheet. Control signals and operating conditions for conventional $\overline{DA}$ addressing are summarized in Table 1.

Fast Select Operations (MS = 0)

When MS is tied low, the fast select addressing mode is invoked; in this mode, the 8X450 address lines are driven by an external source and the 8X450 data lines are tied to the 8X305 $\overline{IV}$ bus. The $\overline{CS}$ input is treated exactly the same as in conventional $\overline{DA}$ addressing; examples of single and multiple RAM configurations are shown at the rear of this data sheet. Control signals and operating conditions for fast select addressing are summarized in Table 2.

PIN CONFIGURATION AND DECRIPTIONS

PIN NO.	IDENTIFIER	FUNCTION
1 – 8	D7 – D0	Bidirectional data bus, D0 is LSB
9	$\overline{CS}$	Chip Select, active low
10	$\overline{ME}$	Master Enable, active low
11	WC	Write Control, active high
12	GND	Ground
13	SC	Select Control, active high
14	MS	Mode Select
15	MCLK	Master Clock, active high
16 – 23	A0 – A7	Address lines, A0 is LSB
24	V_{CC}	Power supply

Table 1. Conventional $\overline{DA}$ Select (MS = 1)

SC	WC	$\overline{ME}$	MCLK	FUNCTION
0	0	0	0	Read from RAM
0	0	1	X	None
0	1	0	1	Write data to RAM
0	1	1	1	None
1	0	0	1	Conventional $\overline{DA}$ select (write address to RAM)
1	0	1	1	None

NOTES:

1. Other states are not generated by the 8X305 and are not defined for operation in this mode.
2. Table 1 is valid only if the chip select ($\overline{CS}$) is Low; if $\overline{CS}$ is High, the 8X450 is disabled.

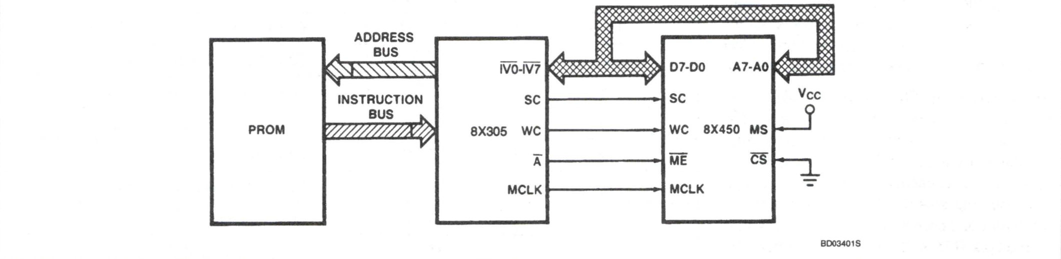
Table 2. Fast Select (MS = 0)

SC	WC	$\overline{ME}$	MCLK	FUNCTION
0	0	0	0	Read from RAM
0	0	1	0	None
X	X	1	1	Write address to RAM
0	1	0	1	Write data to RAM

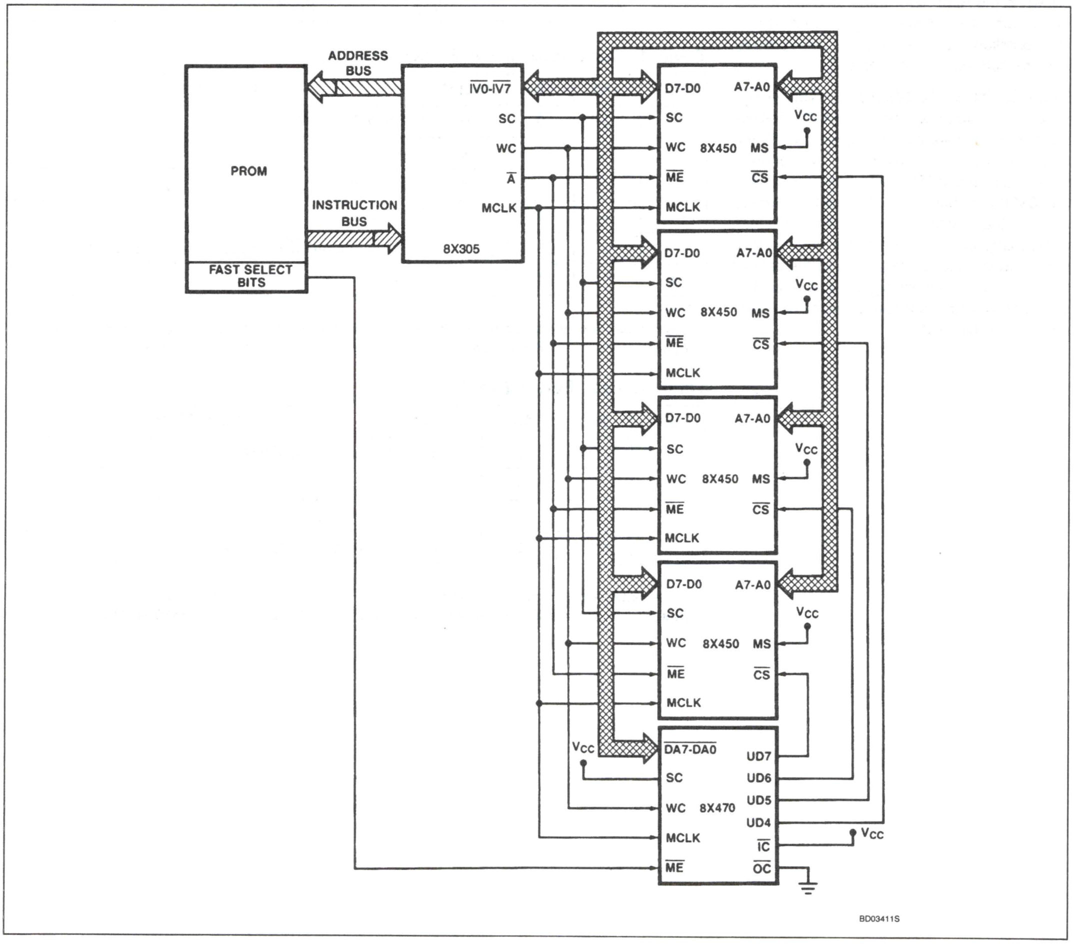
NOTES:

1. Other states are not generated by the 8X305 and are not defined for operation in this mode.
2. Table 2 is valid only if the chip select ($\overline{CS}$) is Low. If $\overline{CS}$ is High, the 8X450 is disabled.

SINGLE RAM CONFIGURATION (CONVENTIONAL $\overline{DA}$ SELECT)

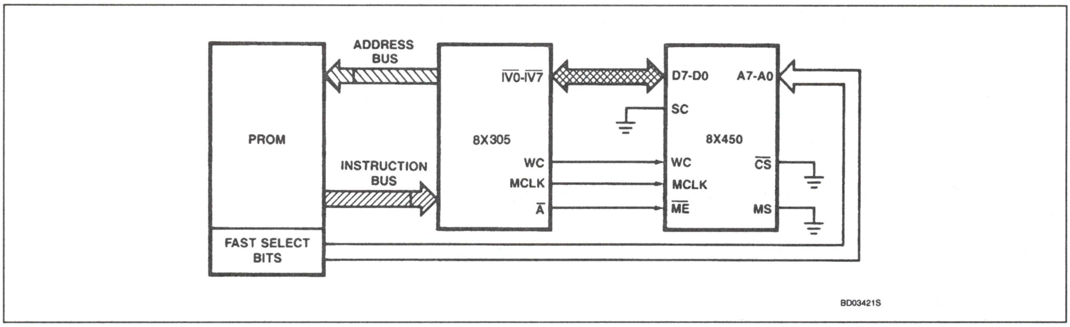

PROM
ADDRESS BUS
INSTRUCTION BUS
$\overline{IV0}$-$\overline{IV7}$
SC
8X305 WC
$\overline{A}$
MCLK
D7-D0
A7-A0
SC
WC 8X450 MS
$\overline{ME}$
$\overline{CS}$
MCLK
V_{CC}
BD03401S

MULTIPLE RAM CONFIGURATION (CONVENTIONAL $\overline{DA}$ SELECT)

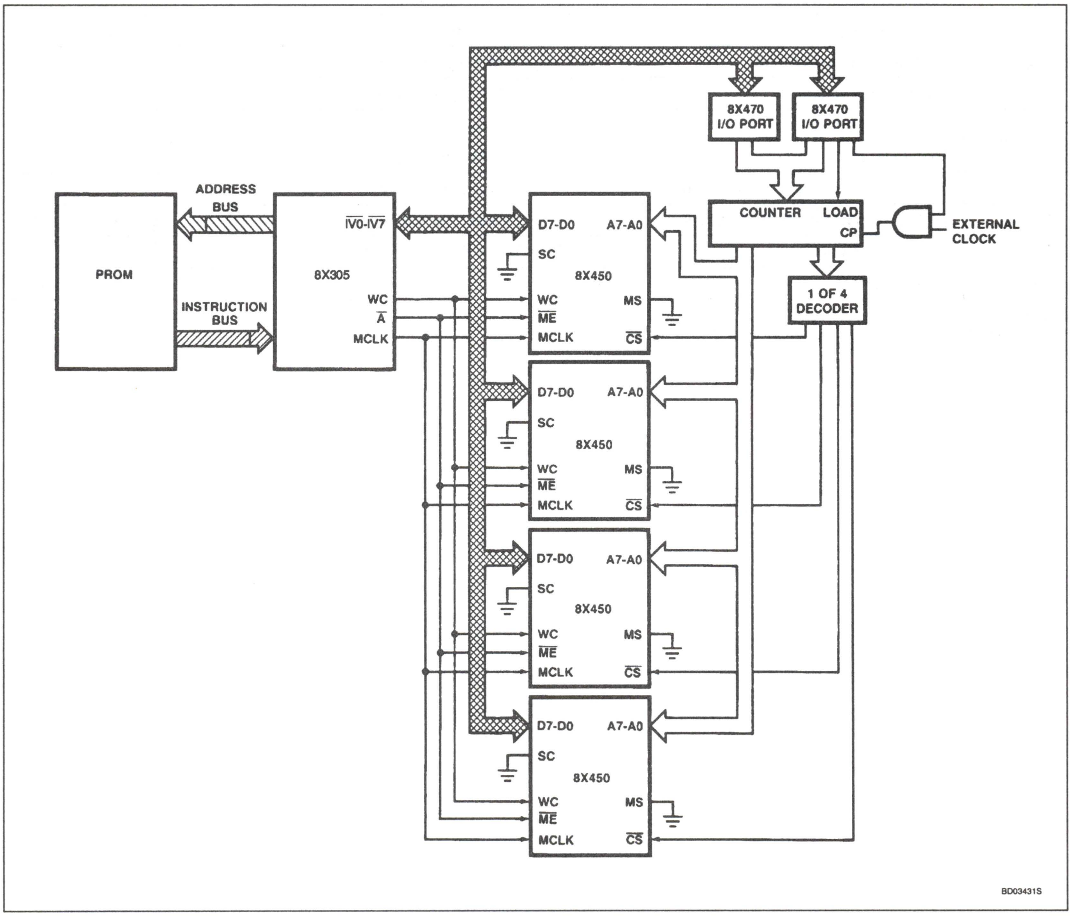

ADDRESS BUS
PROM
INSTRUCTION BUS
FAST SELECT BITS
$\overline{IV0}$-$\overline{IV7}$
SC
WC
$\overline{A}$
MCLK
8X305
D7-D0
A7-A0
SC
WC 8X450 MS
$\overline{ME}$
$\overline{CS}$
MCLK
V_{CC}
DA7-DA0
UD7
UD6
UD5
UD4
WC 8X470
$\overline{IC}$
$\overline{OC}$
BD03411S

RAM 8X450

SINGLE RAM CONFIGURATION (FAST SELECT)


PROM
FAST SELECT BITS
ADDRESS BUS
INSTRUCTION BUS
8X305
IV0-IV7
WC
MCLK
A
D7-D0
A7-A0
SC
8X450
WC
CS
MCLK
ME
MS
BD03421S

MULTIPLE RAM CONFIGURATION (FAST SELECT)


PROM
ADDRESS BUS
INSTRUCTION BUS
8X305
IV0-IV7
WC
A
MCLK
8X470 I/O PORT
8X470 I/O PORT
COUNTER
LOAD
CP
EXTERNAL CLOCK
1 OF 4 DECODER
D7-D0
A7-A0
SC
8X450
WC
ME
MCLK
MS
CS
BD03431S

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_O	Output off-state voltage	+5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS ($V_{CC} = 5V\ +5\%,\ 0°C \leqslant T_A \leqslant 70°C$)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{CC}	Supply voltage		4.75	5	5.25	V
V_{IL}	Low-level input voltage				0.8	V
V_{IH}	High-level input voltage		2			V
V_{OL}	Low-level output voltage	V_{CC} = Min, I_{OL} = 16mA			0.5	V
V_{OH}	High-level output voltage	V_{CC} = Min, I_{OH} = −3mA	2.7			V
V_{IC}	Input clamp voltage	V_{CC} = Min, I_{IN} = −10mA			−1.5	V
I_{IL}	Low-level input current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{IH}	High-level input current	V_{CC} = Max, V_I = 2.7V			20	µA
I_{OZL}	High-Z output current (Low Level)	V_{CC} = Max, V_O = 0.4V			−0.4	mA
I_{OZH}	High-Z output current (High Level)	V_{CC} = Max, V_O = 2.7V			50	µA
I_{OS}	Short circuit output current	V_{CC} = Max + 0.5, V_O = 0.5V	−30		−140	mA
I_{CC}	Supply current	V_{CC} = Max			220	mA

NOTE:
*Not more than one output should be tested at a time.

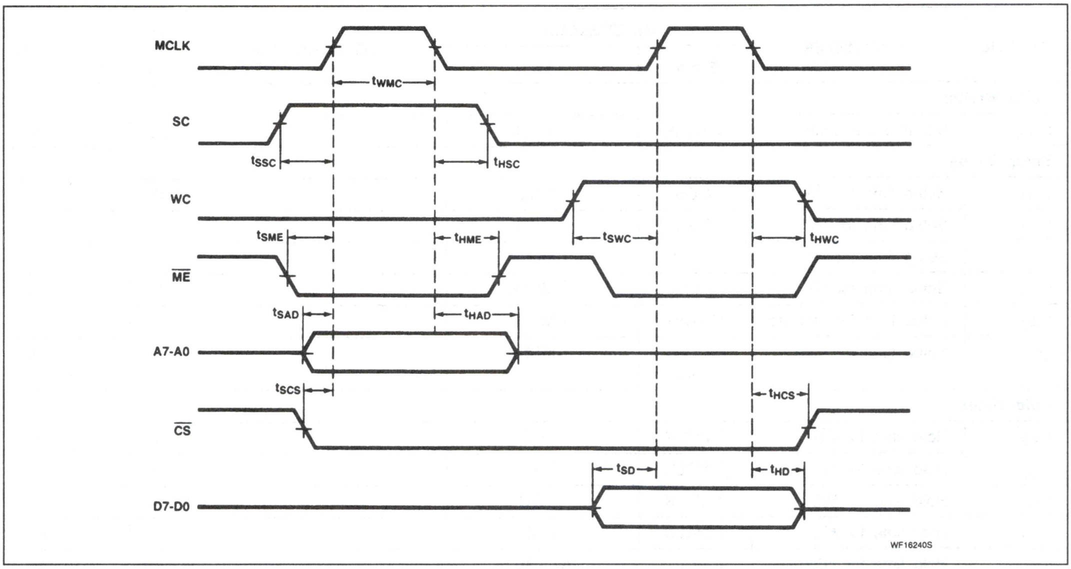
AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V +5%, 0°C $\leqslant T_A \leqslant$ 70°C)

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS		UNIT
		From	To		Min	Max	
Pulse Widths							
t_{WMC}	MCLK pulse width	↑ MCLK	↓ MCLK		35		ns
Setup Times							
t_{SCS}	Setup time for $\overline{CS}$	↓ $\overline{CS}$	↑ MCLK		0		ns
t_{SSC}	Setup time for SC	↑ SC	↑ MCLK		15		ns
t_{SWC}	Setup time for WC	↑ WC	↑ MCLK		15		ns
t_{SME}	Setup time for $\overline{ME}$	↑ $\overline{ME}$	↑ MCLK		0		ns
t_{SAD}	Setup time for address	A7 – A0	↑ MCLK		0		ns
t_{SD}	Setup time for data	$\overline{DA}$ Bus Active	↑ MCLK		0		ns
Hold Times							
$t_{HCS}{}^3$	Hold time for $\overline{CS}$	↓ MCLK	↑ $\overline{CS}$		0		ns
t_{HSC}	Hold time for SC	↓ MCLK	↓ SC		0		ns
t_{HWC}	Hold time for WC	↓ MCLK	↓ WC		0		ns
t_{HME}	Hold time for $\overline{ME}$	↓ MCLK	↑ $\overline{ME}$		0		ns
t_{HAD}	Hold time for address	↓ MCLK	A7 – A0		10		ns
t_{HD}	Hold time for data	↓ MCLK	$\overline{DA}$ Bus 3-Stated		5		ns
Enable Times							
t_{ECS}	Enable time for $\overline{CS}$	↓ $\overline{CS}$	$\overline{DA}$ Bus Active	$\overline{ME}$, SC, WC, MCLK = Low		30	ns
t_{EME}	Enable time for $\overline{ME}$	↓ $\overline{ME}$	$\overline{DA}$ Bus Active	$\overline{CS}$, SC, WC, MCLK = Low		30	ns
t_{ESC}	Enable time for SC	↓ SC	$\overline{DA}$ Bus Active	$\overline{CS}$, $\overline{ME}$, WC, MCLK = Low		30	ns
t_{EWC}	Enable time for WC	↓ WC	$\overline{DA}$ Bus Active	$\overline{CS}$, $\overline{ME}$, SC, MCLK = Low		30	ns
Disable Times							
t_{DCS}	Disable time for $\overline{CS}$	↑ $\overline{CS}$	$\overline{DA}$ Bus 3-Stated	$\overline{ME}$, SC, WC, MCLK = Low		40	ns
t_{DME}	Disable time for $\overline{ME}$	↑ $\overline{ME}$	$\overline{DA}$ Bus 3-Stated	$\overline{CS}$, SC, WC, MCLK = Low		40	ns
t_{DSC}	Disable time for SC	↑ SC	$\overline{DA}$ Bus 3-Stated	$\overline{CS}$, $\overline{ME}$, WC, MCLK = Low		40	ns
t_{DWC}	Disable time for WC	↑ WC	$\overline{DA}$ Bus 3-Stated	$\overline{CS}$, $\overline{ME}$, SC, MCLK = Low		40	ns

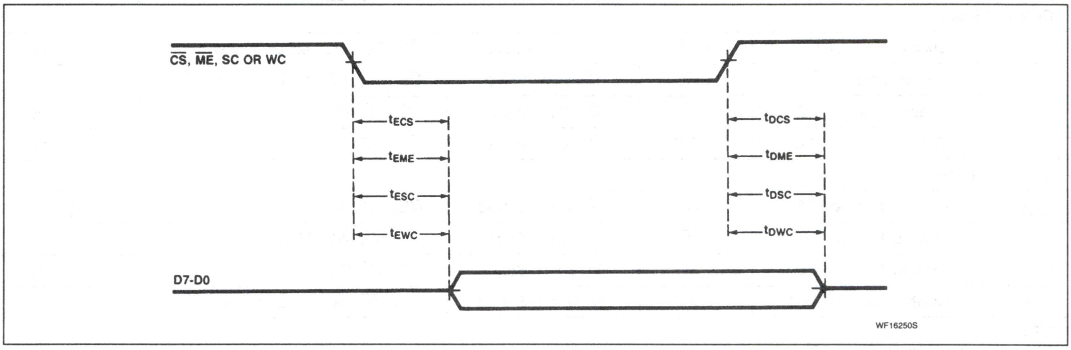
NOTES:

1. All inputs swing between 0V and 3V.
2. All outputs are measured at 1.5V with loading specified in TEST CIRCUIT.
3. $\overline{CS}$ must be active for at least one complete 8X305 instruction cycle.

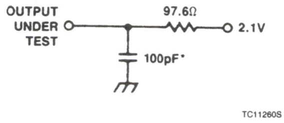
8X450 TIMING


MCLK
SC
WC
ME
A7-A0
CS
D7-D0
tWMC
tSSC
tHSC
tSME
tHME
tSWC
tHWC
tSAD
tHAD
tSCS
tHCS
tSD
tHD
WF16240S

ENABLE AND DISABLE TIMING DIAGRAM


CS, ME, SC OR WC
tECS
tEME
tESC
tEWC
tDCS
tDME
tDSC
tDWC
D7-D0
WF16250S

TEST CIRCUIT


OUTPUT UNDER TEST
97.6Ω
2.1V
100pF*
TC11260S

NOTE:
*Includes test fixture capacitance.

Signetics

8X470
I/O Port

Product Specification

Microprocessor Products

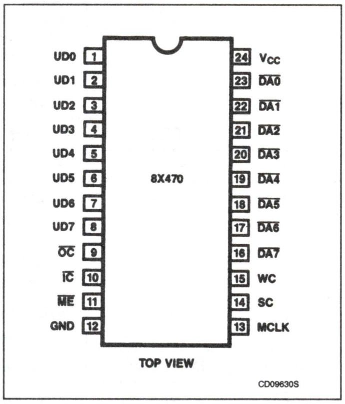
DESCRIPTION

The 8X470 I/O Port (Figure 1) is a fuse-programmable device designed to interface a user system with a Signetics 8X401 Microcontroller. The 8X470 has three internal registers — a data register and two status registers. The data register is shared between the two systems, while each system has its own dedicated status register. The 8X470 can be programmed to provide a bidirectional or unidirectional user interface with the user data entry being either synchronous or asynchronous with the 8X401.

FEATURES

- **Two independent 8-bit busses**
- **Fast polling**
- **Directly bus-compatible with the 8X401, 8X305 Microcontrollers**
- **256 Data Addresses**
- **Separate Controller and User status registers**
- **16 Status Addresses**
- **Positionable Controller Status Register output**
- **Bidirectional or unidirectional I/O capability**
- **Synchronous or asynchronous user data entry**
- **Conventional $\overline{DA}$ or Fast select addressing techniques**
- **24-pin slimline package**
- **Single +5V supply**

PIN CONFIGURATION


UD0 1
UD1 2
UD2 3
UD3 4
UD4 5
UD5 6
UD6 7
UD7 8
OC 9
IC 10
ME 11
GND 12
24 Vcc
23 DA0
22 DA1
21 DA2
20 DA3
19 DA4
18 DA5
17 DA6
16 DA7
15 WC
14 SC
13 MCLK
8X470
TOP VIEW
CD09630S

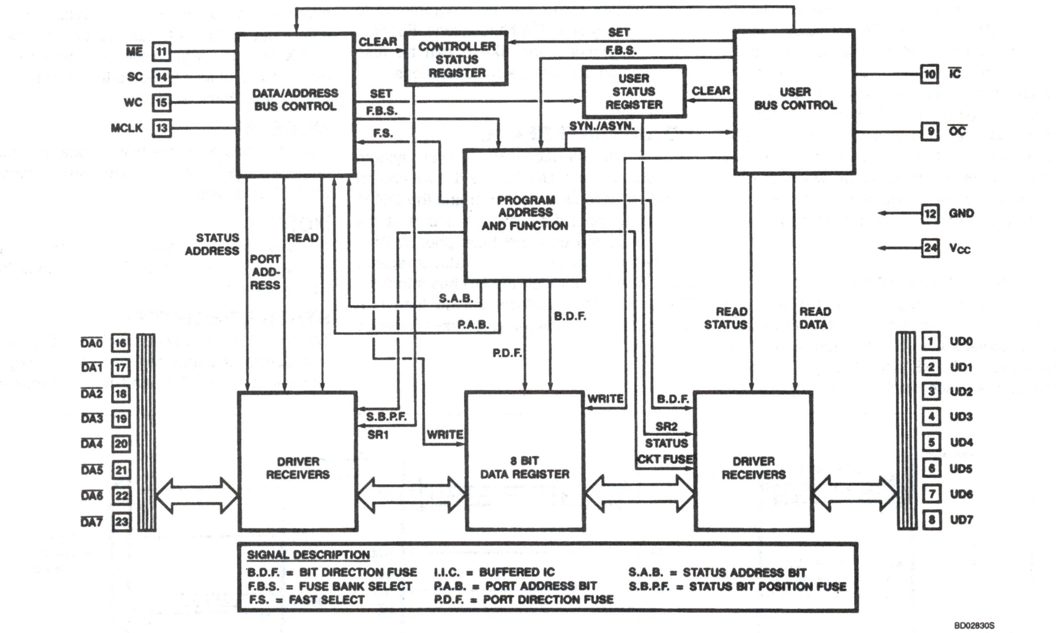

ME 11
SC 14
WC 15
MCLK 13
DATA/ADDRESS BUS CONTROL
CLEAR
CONTROLLER STATUS REGISTER
SET
F.B.S.
USER STATUS REGISTER
CLEAR
USER BUS CONTROL
SET
F.B.S.
F.S.
SYN./ASYN.
10 IC
9 OC
PROGRAM ADDRESS AND FUNCTION
12 GND
24 Vcc
STATUS ADDRESS
READ
PORT ADDRESS
S.A.B.
P.A.B.
B.D.F.
P.D.F.
READ STATUS
READ DATA
DA0 16
DA1 17
DA2 18
DA3 19
DA4 20
DA5 21
DA6 22
DA7 23
DRIVER RECEIVERS
S.B.P.F.
SR1
WRITE
8 BIT DATA REGISTER
WRITE
B.D.F.
SR2 STATUS CKT FUSE
DRIVER RECEIVERS
1 UD0
2 UD1
3 UD2
4 UD3
5 UD4
6 UD5
7 UD6
8 UD7
SIGNAL DESCRIPTION
B.D.F. = BIT DIRECTION FUSE
F.B.S. = FUSE BANK SELECT
F.S. = FAST SELECT
I.I.C. = BUFFERED IC
P.A.B. = PORT ADDRESS BIT
P.D.F. = PORT DIRECTION FUSE
S.A.B. = STATUS ADDRESS BIT
S.B.P.F. = STATUS BIT POSITION FUSE
BD02830S

Figure 1. 8X470 Block Diagram

PIN DESCRIPTION

PIN	IDENTIFIER	FUNCTION
1 – 8	UD0 – UD7	User port data lines. Active high.
9	$\overline{OC}$	**Output Control:** An active low signal which controls reading into the I/O port from the UD bus.
10	$\overline{IC}$	**Input Control:** An active low signal which controls writing into the I/O port from the UD bus.
11	$\overline{ME}$	**Master Enable:** The $\overline{ME}$ pin depends upon the addressing mode for its function. Conventional $\overline{DA}$ select: $\overline{ME}$ is an active low signal which allows read or write or address operations from the $\overline{DA}$ bus. Fast select: $\overline{ME}$ is an active high signal which allows the controller status register to be read on the $\overline{DA}$ bus.
12	GND	**Ground.**
13	MCLK	**Master Clock:** Input from the 8X401.
14	SC	**Select Command:** The SC pin depends upon the addressing mode for its function. Conventional $\overline{DA}$ select: An active high signal which compares the 8X401 $\overline{DA}$ bus and the address. An address match selects the 8X470. Fast select: SC is an active high signal which outputs the contents of the data register on the $\overline{DA}$ bus while clearing the control status register.
15	WC	**Write Command:** An active high signal which controls the writing of data into the I/O port from the 8X401 $\overline{DA}$ bus.
16 – 23	$\overline{DA7}$ – $\overline{DA0}$	Controller Data/Address ($\overline{DA}$) lines. Active low.
24	V_{CC}	+5V power supply.

In a typical system configuration (Figure 2), the 8X470 connects to the 8X401 Microcontroller via the Data/Address ($\overline{DA}$) bus; the 8X470 is linked to the user system via the User Data (UD) bus. To facilitate interbus communications and to implement handshaking operations, a dedicated status register is assigned to each bus. The status registers can be polled at full speed to optimize throughput and to prevent errors in data transmissions. The 8X470 can be addressed from the $\overline{DA}$ bus in either of two ways — by the conventional microcontroller software-select cycle or by fast select methods.

$\overline{DA}$ BUS INTERFACE

The 8X470 interfaces to the 8X401 Microcontroller through an 8-bit inverted three-state Data/Address ($\overline{DA}$) bus and four control signals (SC, WC, $\overline{ME}$ and MCLK). The status registers are accessible through the $\overline{DA}$ bus interface. For conventional select operation, both the data and controller status register addresses can be fuse programmed. Also, the controller status can be programmed to appear at any or all $\overline{DA}$ bit locations. For time critical operations, the 8X470 can be fuse programmed for fast select operations.

UD BUS INTERFACE

The 8X470 interfaces to a user system through an 8-bit UD bus and two control signals ($\overline{IC}$ and $\overline{OC}$). If the user and the 8X401 attempt to input data to the I/O port at the same time, the user will have priority. Both the data register and the user status register are accessible through the UD bus interface. The 8X470 UD interface can be fuse programmed for synchronous or asynchronous data entry.

Data Entry

Data entry into the 8X470 from the user bus can be fuse programmed to be either synchronous or asynchronous. Synchronous data input to the data latches occurs when MCLK is high. Asynchronous data input is independent of MCLK. For further details, see the $\overline{IC}$ fuse bank section.

MODE A

The user data lines are bidirectional or input only. See the $\overline{IC}$ and SC fuse bank sections for further details.

MODE B

The user data lines are output or input only. See the $\overline{IC}$ and SC fuse bank sections for further details.

STATUS REGISTERS

The 8X470 has two status registers; the User and Controller status registers. These registers can be used for handshaking between

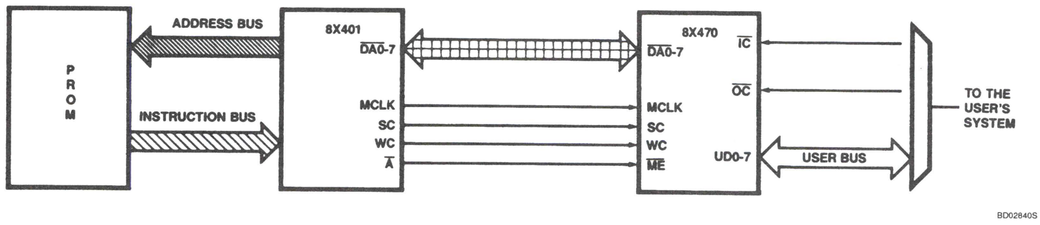


Figure 2. Typical System Configuration

the 8X401 and the User Data Bus. A register is set by a write from one bus and cleared by reading the data register from the opposite bus.

User Status Register

The User Status register is set by any write from the $\overline{DA}$ bus and cleared by reading the data register from the UD bus. The user status register is available on UD7 only. The output of the user status is controlled by $\overline{IC}$ and $\overline{OC}$ user port control lines. See the $\overline{IC}$ bank for further details.

Controller Status Register

The Controller Status register is set by any write from the UD bus and cleared by reading the data register from the $\overline{DA}$ bus. The Controller Status register can be accessed by either a fast select or a conventional $\overline{DA}$ select. For conventional select methods, the controller status register address can be fuse programmed to be any one of the addresses $F0-FF_{16}$. The programming of this address is discussed in the $\overline{ME}$ and $\overline{IC}$ fuse bank sections. The Controller Status register can also be programmed to appear at any or all $\overline{DA}$ bit locations, enhancing the polling process of multiple 8X470 I/O ports.

POLLING

The Controller Status register address and output location can be fuse programmed to allow multiple 8X470 I/O ports to be polled in a single cycle. The 8X470 I/O ports to be polled are programmed with a common Controller Status address, with each status register output available on a different $\overline{DA}$ line to prevent a bus contention. The 8X401 Microcontroller can poll a maximum of eight 8X470 I/O ports in a single cycle, obtaining up to eight bits of status information. The 8X401 Microcontroller's subfield manipulation capability can then be used to isolate a single 8X470 status bit, simplifying the polling procedure and improving the overall system throughput. Figure 3 shows a typical configuration of eight 8X470 I/O ports polled in a single cycle.

ADDRESSING MODES

The 8X470 can be addressed by a microcontroller through either Fast Select or Conventional $\overline{DA}$ select techniques.

Conventional $\overline{DA}$ Select

The 8X470 must be preselected through a conventional $\overline{DA}$ select cycle before either the data or status registers can be accessed by a Microcontroller. During a conventional $\overline{DA}$ select cycle, the 8X470 compares the address on the $\overline{DA}$ bus to the data and status addresses. If either address matches the $\overline{DA}$ address, the appropriate 8X470 latch is set, enabling the port for a data transfer or a polling cycle. Table 1 shows the conventional select operation of the 8X470 $\overline{DA}$ interface.

Fast Select

A fast select cycle is ideal for time critical applications which require port selection and data transfer in a single clock cycle. Additional bits in the 8X401 instruction word are used to execute a fast select cycle. It should be noted that the 8X470 SC input is not driven by the 8X401 SC output and could introduce a minor operational limitation. For an I/O address select instruction, the 8X401 will output a $\overline{DA}$ address and assert SC. However, a fast select configured 8X470 will also drive the $\overline{DA}$ bus, and a bus contention will result. To avoid this contention, the 8X470 should not be fast selected during any I/O address select instruction. Figure 4 shows a typical fast select configuration. Table 2 shows the fast select operation of the 8X470 $\overline{DA}$ bus interface operation.

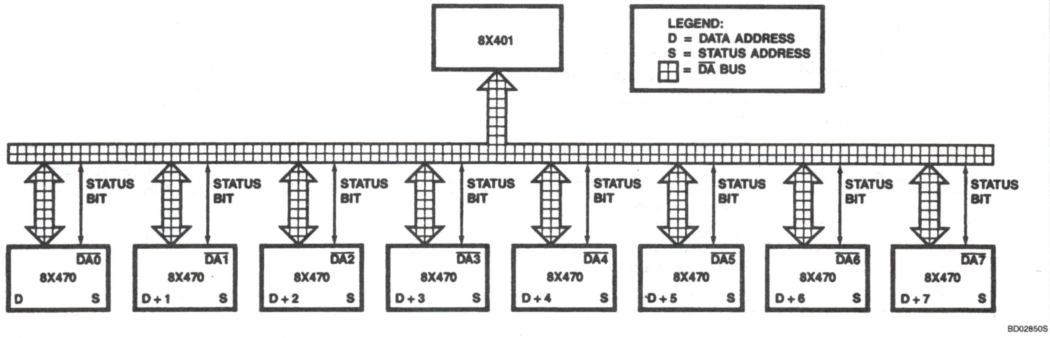


Figure 3. Typical Polling Configuration

I/O Port 8X470

Table 1. Conventional $\overline{\text{DA}}$ Operation

$\overline{\text{ME}}$	SC	WC	MCLK	ADDRESS LATCH	STATUS LATCH	FUNCTION
L	L	L	X	SET	CLEAR	Output data, clear CONTROLLER STATUS
L**	L	H	H	SET	CLEAR	Input 8X401 data, set USER STATUS
L	H	L	H	X	X	Compare enable address
H	X	X	X	X	X	High impedance
L	L	L	X	CLEAR	SET	Output CONTROLLER STATUS
L	L	H	X	CLEAR	SET	High impedance
L	L	H	L	X	X	High impedance
L	H	L	L	SET	X	High impedance
L	L	L	X	CLEAR	CLEAR	High impedance

NOTE:
**If the user and the 8X401 attempt to input data to the I/O port at the same time, the user will overwrite the 8X401 (user priority).

Table 2. Fast Select Operation

$\overline{\text{ME}}$	SC	WC	MCLK	FUNCTION
L	H	H	H	Input 8X401 data, set USER STATUS
L	H	L	X	Output data, clear CONTROLLER STATUS
H	L	L	X	Output CONTROLLER STATUS
H	H	X	X	Inactive
L	L	X	X	Inactive

Table 3. 8X470 Fuse Mnemonics

BANK	PROGRAMMABLE FUSES	PROGRAMMED FUNCTION
SC	UD0 – UD7	Input Only
WC	UD0 – UD7	Controller Data Address (CDA0 – CDA7)
$\overline{\text{ME}}$	UD0 – UD7	Controller Status Location (CSL0 – CLS7)
$\overline{\text{IC}}$	UD0 – UD3	Controller Status Address (CSA0 – CSA3)
$\overline{\text{IC}}$	UD4	Fast Select
$\overline{\text{IC}}$	UD5	Bidirectional Disable
$\overline{\text{IC}}$	UD6	Asynchronous Input

Fuses

The 8X470 has four programmable fuse banks, namely the SC, WC, $\overline{\text{ME}}$ and $\overline{\text{IC}}$ banks. Each bank is selected by its namesake pin, i.e. the SC pin selects the SC fuse bank, the WC pin selects the WC fuse bank, etc. Each bank has eight fuses, except the $\overline{\text{IC}}$ bank, which only has seven. These fuse banks govern different modes of operation, and by selecting the appropriate fuse and fuse bank, the 8X470 can be configured for the modes of operation as mentioned earlier (see Figure 5 for an illustration of the fuse banks). The mnemonics for the fuses are listed in Table 3.

	LSB UD0	UD1	UD2	UD3	UD4	UD5	UD6	MSB UD7
SC	☐	☐	☐	☐	☐	☐	☐	☐
WC	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{\text{ME}}$	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{\text{IC}}$	☐	☐	☐	☐	☐	☐	☐	

DF05790S

Figure 5. Fuse Bank Summary

$\overline{\text{IC}}$ BANK

CSA0 – CSA3

The first four fuses of the $\overline{\text{IC}}$ bank (UD0 – UD3) program the four least significant bits of the Controller Status address (CSA0 – 3). The four most significant bits of the Controller Status address are always a 1. An unprogrammed (unblown) CSA fuse is a logic 1 from the $\overline{\text{DA}}$ bus, and a programmed (blown) CSA fuse is a logic 0 from the $\overline{\text{DA}}$ bus. The Controller Status register can be assigned any one of the addresses F0 – FF (00 – 0F on the $\overline{\text{DA}}$ bus). In an unprogrammed part, the status register will be assigned address F0.

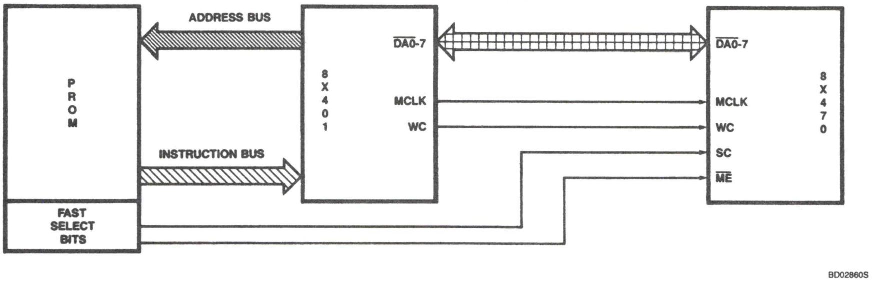


Figure 4. Typical Fast Select Configuration

Fast Select Fuse

The Fast select fuse programs the addressing mode; i.e., Fast or Conventional $\overline{\text{DA}}$ select. In an unprogrammed part, Conventional $\overline{\text{DA}}$ select is selected.

Bidirectional Disable Fuse

The bidirectional disable fuse selects either Mode A or Mode B operation. In an unprogrammed part, Mode A is selected. Modes A and B are defined more fully in the SC fuse bank.

Asynchronous Input Fuse

The asynchronous input fuse selects the user data entry mode; synchronous or asynchronous. In an unprogrammed part, data is entered synchronously with MCLK.

See Table 4 for a summary of the $\overline{\text{IC}}$ bank.

SC Bank

The SC bank has two possible configurations, Mode A and Mode B, as selected by the bidirectional disable fuse. Mode A allows bidirectional and input only UD ports; Mode B allows output only and input only UD ports. Input only UD ports are output only from the $\overline{\text{DA}}$ bus, all other types of UD ports (bidirectional and output only) are bidirectional from the $\overline{\text{DA}}$ bus at the same bit location (see Figure 6).

Mode A

Mode A is selected when the bidirectional disable fuse is unblown, configuring the unprogrammed SC bank for bidirectional operations (see Table 5). The user would then blow the needed input only fuses to disable the bidirectional capability of that UD line, making it input only. This line is input only from the UD bus and output only from the $\overline{\text{DA}}$ bus (see Example 1). Table 6 gives the Mode A operation of the 8X470 user interface.

Table 5. Mode A Bus Operation

INPUT ONLY FUSE "N"	UD "N"	$\overline{\text{DA}}$ "N"
Unblown	B	B
Blown	I	O

I = Input only
O = Output only
B = Bidirectional

Table 4. $\overline{\text{IC}}$ Fuse Bank Summary

UD PIN	FUNCTION	UNBLOWN FUSE	BLOWN FUSE
UD0 – UD3	Controller Status Address	Logic 1 from the $\overline{\text{DA}}$ bus	Logic 0 from the $\overline{\text{DA}}$ bus
UD4	Addressing mode	Conventional $\overline{\text{DA}}$ Select	Fast Select
UD5	Bidirectional disable	Mode A	Mode B
UD6	Data entry	Synchronous	Asynchronous

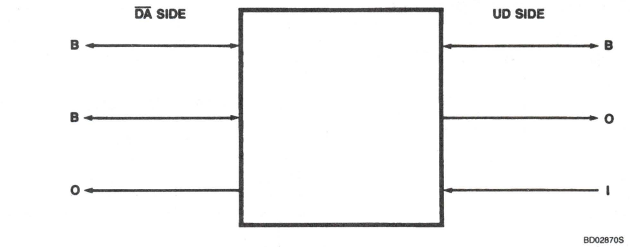


Figure 6. Port Direction Summary

Example 1. Mode A Unprogrammed and Programmed Configurations

	LSB 0	1	2	3	4	5	6	MSB 7
SC	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{\text{WC}}$	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{\text{ME}}$	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{\text{IC}}$	☐	☐	☐	☐	☐	☐	☐	

DF05800S

	LSB 0	1	2	3	4	5	6	MSB 7
SC	■	■	■	■	☐	☐	☐	☐
$\overline{\text{WC}}$	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{\text{ME}}$	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{\text{IC}}$	☐	☐	☐	☐	☐	☐	☐	

DF05860S

BUS SIDE	BIT LOCATION 0	1	2	3	4	5	6	7
UD	B	B	B	B	B	B	B	B
$\overline{\text{DA}}$	B	B	B	B	B	B	B	B

BUS SIDE	BIT LOCATION 0	1	2	3	4	5	6	7
UD	I	I	I	I	B	B	B	B
$\overline{\text{DA}}$	O	O	O	O	B	B	B	B

Table 6. Mode A Operation (See Note)

BIDIRECTIONAL			INPUT ONLY		
$\overline{\text{OC}}$	$\overline{\text{IC}}$	Function	$\overline{\text{OC}}$	$\overline{\text{IC}}$	Function
L	L	High impedance	L	L	High impedance
L	H	Output data, clear user status	L	H	High impedance (UD7 = V_{OH})
H	L	Input user data, set controller status	H	L	Input user data, set controller status
H	H	High impedance (UD7 = User Status)	H	H	High impedance (UD7 = User Status)

NOTE:
This table should only be used if the bidirectional fuse is unblown. For UD lines where the input only fuse is blown, use the input only side; else use the bidirectional side.

Mode B

Mode B is selected when bidirectional disable fuse is blown, configuring the unprogrammed SC bank for output only operation from the UD bus (see Table 7). The user would then blow the needed input only fuses to disable the output only capability of that UD line, making it input only. The line is input only from the UD bus and output only from the $\overline{DA}$ bus (see Example 2). Table 8 gives the Mode B operation of the 8X470 user interface.

Table 7. Mode B Bus Operation

INPUT ONLY FUSE "N"	UD "N"	$\overline{DA}$ "N"
Unblown	O	B
Blown	I	O

I = Input only
O = Output only
B = Bidirectional

WC Bank

The WC fuses (CDA0 – CDA7) are used to program the Data Address for Conventional $\overline{DA}$ select operations. An unprogrammed WC fuse requires a logic 1 at its $\overline{DA}$ bit location to produce an address match. A blown fuse requires a logic 0 at its $\overline{DA}$ bit location to produce an address match. The user can then program the data address to be any one of the addresses 00 – FF. An unprogrammed 8X470 recognizes a $\overline{DA}$ address of FF (00 within the 8X401) as a data address match (see Example 3).

Example 3. Controller Data Register Address

BUS SIDE	LSB 0	1	2	3	4	5	6	MSB 7
UD	☐	■	■	■	■	■	■	☐
$\overline{DA}$	The controller status register output is fixed at $\overline{DA0}$ and $\overline{DA7}$.							

DF06160S

$\overline{ME}$

The $\overline{ME}$ fuse (CSL0 – CSL7) bank positions the output of the Controller Status bit at any unblown UD fuse at the same bit location on the $\overline{DA}$ side. In an unprogrammed part, the Controller Status output will appear on all eight $\overline{DA}$ lines (see Example 4).

Example 2. Mode B Unprogrammed and Programmed Configurations

	LSB 0	1	2	3	4	5	6	MSB 7
SC	☐	☐	☐	☐	☐	☐	☐	☐
WC	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{ME}$	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{IC}$	☐	☐	☐	☐	☐	■	☐	

	LSB 0	1	2	3	4	5	6	MSB 7
SC	■	■	■	■	☐	☐	☐	☐
WC	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{ME}$	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{IC}$	☐	☐	☐	☐	☐	■	☐	

DF06140S

BUS SIDE	BIT LOCATION 0	1	2	3	4	5	6	7
UD	O	O	O	O	O	O	O	O
$\overline{DA}$	B	B	B	B	B	B	B	B

BUS SIDE	BIT LOCATION 0	1	2	3	4	5	6	7
UD	I	I	I	I	O	O	O	O
$\overline{DA}$	O	O	O	O	B	B	B	B

Table 8. Mode B Operation (See Note)

OUTPUT ONLY			INPUT ONLY		
$\overline{OC}$	$\overline{IC}$	Function	$\overline{OC}$	$\overline{IC}$	Function
L	L	Output data, clear User Status	L	L	Input user data, set Controller Status
L	H	Output data, clear User Status	L	H	High impedance (UD7 = V_{OH})
H	L	High impedance (UD7 = V_{OH})	H	L	Input user data, set Controller Status
H	H	High impedance (UD7 = User Status)	H	H	High impedance (UD7 = User Status)

NOTE:
Table 7 should only be used if the bidirectional fuse has been blown. For UD lines where the input only fuse is blown, use the input only side; else use the output only side.

Example 4. Controller Status Output Location

BUS SIDE	LSB 0	1	2	3	4	5	6	MSB 7
UD	■	☐	■	■	☐	☐	■	■
$\overline{DA}$	0	1	0	0	1	1	0	0
	Data Register Address: CD (Hex)*							

DF06170S

PROGRAMMING

Flowchart 1 summarizes the 8X470 programmable features by their function and programming configuration. Flowchart 1 can be used before programming as a guide for mapping the fuses that need to be blown, to configure an 8X470 I/O port for a particular configuration. After the features have been mapped, the programming procedure can begin. Each bank is programmed in a similar manner, with one exception, the $\overline{IC}$ fuse bank. The $\overline{IC}$ fuse bank has two added considerations; when being programmed, any input to UD7 should be removed, and the fuse at UD7 should not be programmed. Fuses can be verified when all programming as been completed or just after a fuse has been programmed. During verification, a blown fuse acts as a high impedance and an unblown fuse acts as a VOL.

Thus, programming the 8X470 I/O port can be broken down into three stages:

(1) Pre-Programming
  a. Decide what features the 8X470 is to have.
  b. Use Flowchart 1 to determine which fuses need to be blown.
  c. Map the fuses to be blown onto a diagram, like the one shown below, by darkening the squares where fuses are to be blown.

(2) Programming
  Use "map" to blow fuses using the programming procedure.

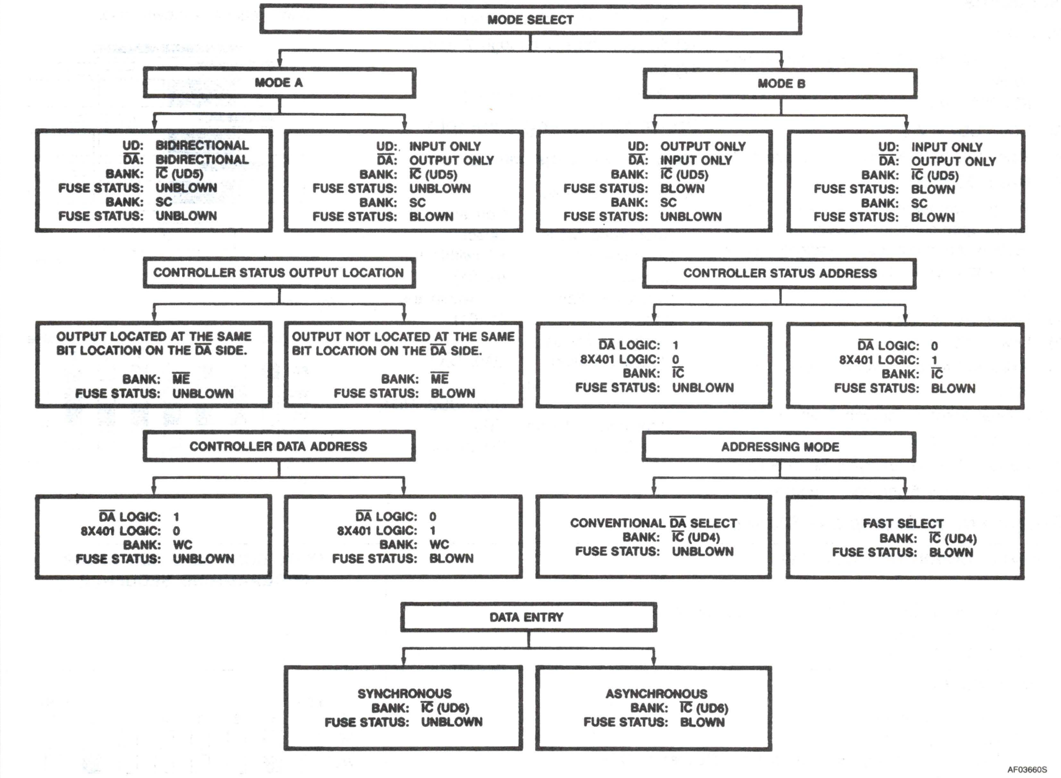
(3) Verify
  Verify that the fuses blown correspond to the map, using the verification procedure.

For an example, see Programming Example.

	UD0	UD1	UD2	UD3	UE4	UD5	UD6	UD7
SC	☐	☐	☐	☐	☐	☐	☐	☐
WC	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{ME}$	☐	☐	☐	☐	☐	☐	☐	☐
$\overline{IC}$	☐	☐	☐	☐	☐	☐	☐	

DF06180S

I/O Port 8X470


MODE SELECT
MODE A
MODE B
UD: BIDIRECTIONAL
$\overline{DA}$: BIDIRECTIONAL
BANK: $\overline{IC}$ (UD5)
FUSE STATUS: UNBLOWN
BANK: SC
FUSE STATUS: UNBLOWN
UD: INPUT ONLY
$\overline{DA}$: OUTPUT ONLY
BANK: $\overline{IC}$ (UD5)
FUSE STATUS: UNBLOWN
BANK: SC
FUSE STATUS: BLOWN
UD: OUTPUT ONLY
$\overline{DA}$: INPUT ONLY
BANK: $\overline{IC}$ (UD5)
FUSE STATUS: BLOWN
BANK: SC
FUSE STATUS: UNBLOWN
UD: INPUT ONLY
$\overline{DA}$: OUTPUT ONLY
BANK: $\overline{IC}$ (UD5)
FUSE STATUS: BLOWN
BANK: SC
FUSE STATUS: BLOWN
CONTROLLER STATUS OUTPUT LOCATION
CONTROLLER STATUS ADDRESS
OUTPUT LOCATED AT THE SAME BIT LOCATION ON THE $\overline{DA}$ SIDE.
BANK: $\overline{ME}$
FUSE STATUS: UNBLOWN
OUTPUT NOT LOCATED AT THE SAME BIT LOCATION ON THE $\overline{DA}$ SIDE.
BANK: $\overline{ME}$
FUSE STATUS: BLOWN
$\overline{DA}$ LOGIC: 1
8X401 LOGIC: 0
BANK: $\overline{IC}$
FUSE STATUS: UNBLOWN
$\overline{DA}$ LOGIC: 0
8X401 LOGIC: 1
BANK: $\overline{IC}$
FUSE STATUS: BLOWN
CONTROLLER DATA ADDRESS
ADDRESSING MODE
$\overline{DA}$ LOGIC: 1
8X401 LOGIC: 0
BANK: WC
FUSE STATUS: UNBLOWN
$\overline{DA}$ LOGIC: 0
8X401 LOGIC: 1
BANK: WC
FUSE STATUS: BLOWN
CONVENTIONAL $\overline{DA}$ SELECT
BANK: $\overline{IC}$ (UD4)
FUSE STATUS: UNBLOWN
FAST SELECT
BANK: $\overline{IC}$ (UD4)
FUSE STATUS: BLOWN
DATA ENTRY
SYNCHRONOUS
BANK: $\overline{IC}$ (UD6)
FUSE STATUS: UNBLOWN
ASYNCHRONOUS
BANK: $\overline{IC}$ (UD6)
FUSE STATUS: BLOWN
AF03660S

Flowchart 1.

Programming Sequence

$V_f = 10V$

1) $V_{CC} = 5V$, $\overline{ME}$, SC, $\overline{IC} = V_{OL}$ (initial conditions).
2) $\overline{IC} = \overline{OC} = V_{OL}$.
3) UD0 – UD7 = V_{OL}.
4) Raise WC to V_f (programming data address).
5) Raise V_{CC} to V_f.
6) Raise UD lines 0 – 7 one at a time where a fuse is to be blown.
7) Lower V_{CC} to 5V.
8) Raise $\overline{OC}$ to V_f.
9) Verify blown fuses on UD0 – UD7. These act as open collector outputs during verification. An intact fuse is indicated by a V_{OL}, a blown fuse by a high impedance.
10) Lower $\overline{OC}$ to V_{OL}.
11) Lower WC to V_{OL}.
12) Raise $\overline{ME}$ to V_f (programming status bit position).
13) Repeat steps 5 through 10 above.
14) Lower $\overline{ME}$ to V_{OL}.
15) Raise SC to V_f (programming data bit direction).
16) Repeat steps 5 through 10 above.
17) Lower SC to V_{OL}.
18) Remove any input from UD7.
19) Raise $\overline{OC}$ to V_{OH}.
20) Raise $\overline{IC}$ to V_f (programming status address and miscellaneous functions).
21) Repeat steps 5 through 10 above, but do not raise UD7 in step 6.
22) Lower $\overline{IC}$ to V_{OL}.
23) Lower $\overline{OC}$ to V_{OL}.
24) Remove power ($V_{CC} = 0V$).

Programming Verification Sequence

V_f = 10V

1) V_{CC} = 5V, $\overline{\text{ME}}$, SC, $\overline{\text{IC}}$ = V_{OL} (initial conditions).
2) $\overline{\text{IC}}$ = $\overline{\text{OC}}$ = V_{OL}.
3) UD0 – UD7 = V_{OL}.
4) Raise WC to V_f (verifying data address).
5) Raise $\overline{\text{OC}}$ to V_f.
6) Verify blown fuses on UD0 – UD7. These act as open collector outputs during verification. An intact fuse is indicated by a V_{OL} and a blown fuse by a high impedance.
7) Lower $\overline{\text{OC}}$ to V_{OL}.
8) Lower WC to V_{OL}.
9) Raise $\overline{\text{ME}}$ to V_f (verifying status bit position).
10) Repeat steps 5 through 7 above.
11) Lower $\overline{\text{ME}}$ to V_{OL}.
12) Raise SC to V_f (verifying bit direction).
13) Repeat steps 5 through 7 above.
14) Lower SC to V_{OL}.
15) Raise $\overline{\text{IC}}$ to V_f (verifying status address and miscellaneous functions).
16) Repeat steps 5 through 7 above.
17) Lower $\overline{\text{IC}}$ to V_{OL}.
18) Remove power (V_{CC} = 0V).

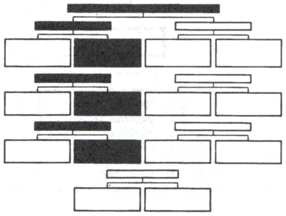
Pre-programming Example

Stage 1 — Pre-programming

(a) Features Desired

Addressing Mode:	Conventional $\overline{\text{DA}}$ select
Controller Data address:	42 (within the 8X401)
Controller Status address:	F0 (within the 8X401)
Controller Status Output Location:	$\overline{\text{DA0}}$, $\overline{\text{DA7}}$
Mode of Operation:	A
Input only UD lines:	UD0, UD2, UD4, UD6
Data Entry:	Synchronous

(b)The shaded areas of Flowchart 1 indicate the selections from pat 'a' which require a blown fuse.



DF05810S

(c) Map

	LSB 0	1	2	3	4	5	6	MSB 7
SC	■	☐	■	☐	■	☐	■	☐
WC	☐	■	☐	☐	☐	☐	■	☐
$\overline{\text{ME}}$	☐	■	■	■	■	■	■	☐
$\overline{\text{IC}}$	☐	☐	☐	☐	☐	☐	☐	

DF05820S

Programming Example — Stages 2 and 3

INITIAL CONDITIONS (V_f = 10V)	**8X470 CONFIGURATION AFTER PROGRAMMING SEQUENCE**
1. V_{CC} = 5V, $\overline{\text{ME}}$, SC, $\overline{\text{IC}}$ = V_{OL} (initial conditions). 2. $\overline{\text{IC}}$ = $\overline{\text{OC}}$ = V_{OL}. 3. UD0 – UD7 = V_{OL}.	
DATA ADDRESS 4. Raise WC to V_f. 5. Raise V_{CC} to V_f. 6. Raise UD1 to V_{OH} to blow a fuse. Repeat of UD6. 7. Lower V_{CC} to 5V. 8. Raise $\overline{\text{OC}}$ to V_f. 9. Verify that the fuses at UD1 and UD6 are blown. A blown fuse acts like a high impedance and an intact fuse acts like a V_{OL}. 10. Lower $\overline{\text{OC}}$ to V_{OL}. 11. Lower WC to V_{OL}.	LSB MSB UD: 0 1 2 3 4 5 6 7 SC ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ WC ☐ ■ ☐ ☐ ☐ ☐ ■ ☐ $\overline{\text{ME}}$ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ $\overline{\text{IC}}$ ☐ ☐ ☐ ☐ ☐ ☐ ☐
CONTROLLER OUTPUT LOCATION 12. Raise $\overline{\text{ME}}$ to V_f. 13. Repeat steps 5 – 10 above for UD1 – UD6. 14. Lower $\overline{\text{ME}}$ to V_{OL}.	LSB MSB UD: 0 1 2 3 4 5 6 7 SC ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ WC ☐ ■ ☐ ☐ ☐ ☐ ■ ☐ $\overline{\text{ME}}$ ☐ ■ ■ ■ ■ ■ ■ ☐ $\overline{\text{IC}}$ ☐ ☐ ☐ ☐ ☐ ☐ ☐
DATA BIT DIRECTION 15. Raise SC to V_f. 16. Repeat steps 5 – 10 above for UD0, UD2, UD4, UD6. 17. Lower SC to V_{OL}.	LSB MSB UD: 0 1 2 3 4 5 6 7 SC ■ ☐ ■ ☐ ■ ☐ ■ ☐ WC ☐ ■ ☐ ☐ ☐ ☐ ■ ☐ $\overline{\text{ME}}$ ☐ ■ ■ ■ ■ ■ ■ ☐ $\overline{\text{IC}}$ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Table 9. Programming Examples

CONFIGURATION LSB MSB UD Bits: 0 1 2 3 4 5 6 7	INPUT/OUTPUT CAPABILITY Bit Location LSB MSB 0 1 2 3 4 5 6 7	DATA ENTRY	ADDRESSING MODE	CONTROLLER STATUS POSITION	CONTROLLER STATUS ADDRESS (within the 8x401)	CONTROLLER DATA ADDRESS (within the 8x401)	MODE OF OPER-ATION
Example 1 SC ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ WC ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ $\overline{ME}$ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ $\overline{IC}$ ☐ ☐ ☐ ☐ ☐ ☐ ☐	UD: B B B B B B B B $\overline{DA}$: B B B B B B B B	Synchronous	Conventional $\overline{DA}$ Select	$\overline{DA0}$ – $\overline{DA7}$	F0 (Hex)	00 (Hex)	A
Example 2 SC ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ WC ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ $\overline{ME}$ ☐ ■ ■ ■ ■ ■ ■ ☐ $\overline{IC}$ ☐ ☐ ☐ ☐ ■ ☐ ☐	UD: B B B B B B B B $\overline{DA}$: B B B B B B B B	Synchronous	Fast Select	$\overline{DA0}$, $\overline{DA7}$	X	X	A
Example 3 SC ■ ☐ ■ ☐ ■ ☐ ■ ☐ WC ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ $\overline{ME}$ ☐ ■ ■ ■ ■ ■ ■ ☐ $\overline{IC}$ ☐ ☐ ☐ ☐ ■ ☐ ☐	UD: I B I B I B I B $\overline{DA}$: O B O B O B O B	Synchronous	Fast Select	$\overline{DA0}$, $\overline{DA7}$	X	X	A
Example 4 SC ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ WC ☐ ■ ☐ ☐ ☐ ☐ ■ ☐ $\overline{ME}$ ☐ ■ ■ ■ ■ ■ ■ ☐ $\overline{IC}$ ☐ ☐ ☐ ☐ ☐ ■ ■	UD: O O O O O O O O $\overline{DA}$: B B B B B B B B	X	Conventional $\overline{DA}$ Select	$\overline{DA0}$, $\overline{DA7}$	F0 (Hex)	42 (Hex)	B
Example 5 SC ■ ■ ■ ■ ☐ ☐ ☐ ☐ WC ☐ ■ ☐ ☐ ☐ ☐ ■ ☐ $\overline{ME}$ ☐ ■ ■ ■ ■ ■ ■ ☐ $\overline{IC}$ ☐ ☐ ☐ ☐ ☐ ■ ■	UD: I I I I O O O O $\overline{DA}$: O O O O B B B B	Asynchronous	Conventional $\overline{DA}$ Select	$\overline{DA0}$, $\overline{DA7}$	F0 (Hex)	42 (Hex)	B

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_O	Off-state voltage	+5.5	V_{DC}
T_{STG}	Storage temperature range	-65 to +150	°C

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Plastic DIP 400 Mil Wide	N8X470N

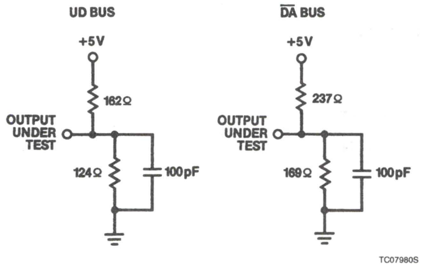
DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V +5%; 0°C ≤ T_A ≤ 70°C)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Low-level input voltage	V_{CC} = Min			0.8	V
V_{IH}	High-level input voltage	V_{CC} = Max	2			V
V_{OL}	Low-level output voltage	V_{CC} = Min I_{OL} = 16 mA from $\overline{\text{DA}}$ bus I_{OL} = 24 mA from UD bus			0.55	V
V_{OH}	High-level output voltage	V_{CC} = Min, I_{OH} = −3mA	2.5			V
V_{CL}	Input clamp voltage	V_{CC} = Min, I_{IN} = −10mA			−1.5	V
I_{IL}	Low-level input current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{IL}	Low-level input current for $\overline{\text{IC}}$, $\overline{\text{ME}}$, SC	V_{CC} = Max, V_I = 0.4V			−0.7	mA
I_{IH}	High-level input current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{OZL}	High-Z output current (low level)	V_{CC} = Max, V_O = 0.4V			−0.4	mA
I_{OZH}	High-Z output current (high level)	V_{CC} = Max, V_O = 2.7V			50	μA
I_{OS}	Short circuit output current	V_{CC} = Min	−30		−140	mA
I_{CC}	Supply current	V_{CC} = Max			220	mA
V_{CC}	Supply voltage		4.75	5	5.25	V

DC FUSING CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_F	Fusing voltage	Applies to V_{CC}, SC, WC, $\overline{\text{ME}}$ and $\overline{\text{IC}}$ only	9.5		11.5	V
I_{FB}	Fuse bank current	$\overline{\text{ME}}$ or SC or WC, $\overline{\text{IC}}$ = 11.5V (Only one bank at a time can equal 11.5V)				mA
I_{CCF}	Fusing current	V_{CC} = 11.5V, $\overline{\text{ME}}$ or SC or WC or $\overline{\text{IC}}$ = 11.5V (Only one bank at a time can equal 11.5 V)				mA

TEST LOADING CIRCUITS



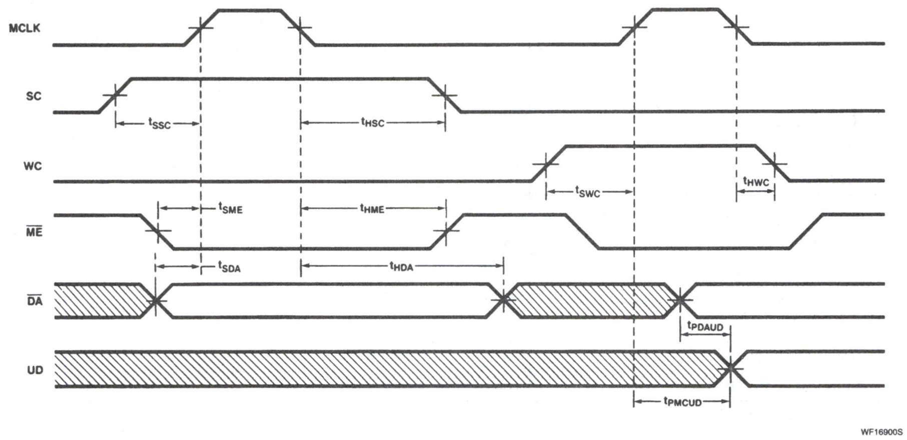
I/O Port 8X470

AC ELECTRICAL CHARACTERISTICS ($V_{CC} = 5V + 5\%$; $0°C \leqslant T_A \leqslant 70°C$) (See Note 1)

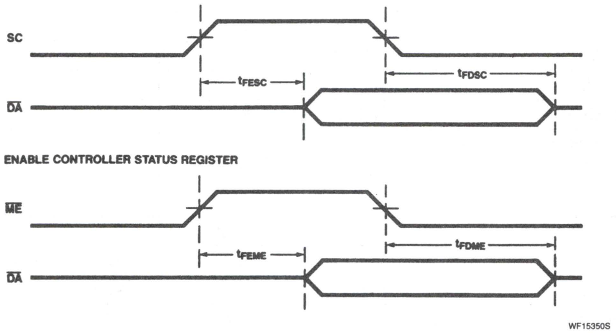
SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS		UNIT
		From	To		Min	Max	
Pulse Widths:							
t_{WMC}	MCLK pulse width			CS = MS = Low	25		ns
t_{WIC}	$\overline{IC}/\overline{OC}$ pulse width			$\overline{ME}$ = High	25		ns
Propagation Delays:							
t_{PUDDA}	Propagation delay from UD to $\overline{DA}$	UD	$\overline{DA}$			35	ns
t_{PICDA}	Propagation delay from $\overline{IC}$ to $\overline{DA}$	$\overline{IC}$ ↓	$\overline{DA}$			35	ns
t_{PMCDA}	Propagation delay from MCLK to $\overline{DA}$	MCLK ↑	$\overline{DA}$			35	ns
t_{PDAUD}	Propagation delay from $\overline{DA}$ to UD	$\overline{DA}$	UD			35	ns
t_{PMCUD}	Propagation delay from MCLK to UD	MCLK ↑	UD			35	ns
Setup Times:							
t_{SSC}	Setup time for SC	SC ↑	MCLK ↑		5		ns
t_{SWC}	Setup time for WC	WC ↑	MCLK ↑		5		ns
t_{SME}	Setup time for $\overline{ME}$	$\overline{ME}$ ↓	MCLK ↑		5		ns
t_{SDA}	Setup time for $\overline{DA}$	$\overline{DA}$	MCLK ↑		0		ns
t_{SIC}	Setup time for $\overline{IC}/\overline{OC}$	$\overline{IC}$ ↓	MCLK ↑		0		ns
t_{SUDIC}	Setup time to enable $\overline{IC}$	UD	$\overline{IC}$ ↓		0		ns
Hold Times:							
t_{HSC}	Hold time for SC	MCLK ↓	SC ↓		0		ns
t_{HWC}	Hold time for WC	MCLK ↓	WC ↓		0		ns
t_{HME}	Hold time for $\overline{ME}$	MCLK ↓	$\overline{ME}$ ↑		0		ns
t_{HIC}	Hold time for $\overline{IC}/\overline{OC}$	MCLK ↓	$\overline{IC}$ ↑		0		ns
t_{HUDIC}	Hold time for between $\overline{IC}$ and UD	$\overline{IC}$ ↑	UD		15		ns
t_{HDA}	Hold time for $\overline{DA}$ bus	MCLK ↓	DA		10		ns
Enable Times:							
t_{EIC}	Enable time for $\overline{IC}$	$\overline{IC}$ ↑	UD output			24	ns
t_{EME}	Enable time for $\overline{ME}$	$\overline{ME}$ ↓	$\overline{DA}$	SC = WC = Low		24	ns
t_{ESC}	Enable time for SC	SC ↓	$\overline{DA}$	WC = $\overline{ME}$ = Low		24	ns
t_{EWC}	Enable time for WC	WC ↓	$\overline{DA}$	SC = $\overline{ME}$ Low		24	ns
t_{ESIC}	Enable time for user status	$\overline{IC}$ ↑	UD7			24	ns
t_{FESC}	Enable time for data in the fast select mode	SC ↑	$\overline{DA}$			24	ns
t_{FEME}	Enable time for the controller status in the fast select mode	$\overline{ME}$ ↑	$\overline{DA}$			24	ns
Disable Times: (see note 2)							
t_{DIC}	Disable time for $\overline{IC}$	$\overline{IC}$ ↓	UD bus 3-Stated			45	ns
t_{DME}	Disable time for $\overline{ME}$	$\overline{ME}$ ↑	$\overline{DA}$ bus 3-Stated			40	ns
t_{DSC}	Disable time for SC	SC ↑	$\overline{DA}$ bus 3-Stated			40	ns
t_{DWC}	Disable time for WC	WC ↑	$\overline{DA}$ bus 3-Stated			40	ns
t_{DSIC}	Disable time for user status	$\overline{IC}$ ↓	UD7 3-Stated			40	ns
t_{FDSC}	Disable time for data in the fast select mode	SC ↓	$\overline{DA}$ bus 3-Stated			40	ns
t_{FDME}	Disable time for controller status in the fast select mode	$\overline{ME}$ ↓	$\overline{DA}$ bus 3-Stated			40	ns

NOTES:
1. Input levels swing from 0V to 3V with Outputs measured at 1.5V.
2. These parameters are measured with a capacitive loading of 100pF.

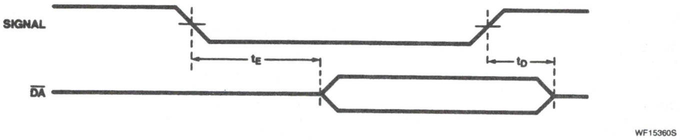
8X470 Timing for the Conventional DA Select Addressing Mode


MCLK
SC
t_{SSC}
t_{HSC}
WC
t_{SWC}
t_{HWC}
$\overline{\text{ME}}$
t_{SME}
t_{HME}
t_{SDA}
t_{HDA}
$\overline{\text{DA}}$
t_{PDAUD}
UD
t_{PMCUD}
WF16900S

8X470 Timing for the Fast Select Addressing Mode


SC
t_{FESC}
t_{FDSC}
$\overline{\text{DA}}$
ENABLE CONTROLLER STATUS REGISTER
$\overline{\text{ME}}$
t_{FEME}
t_{FDME}
$\overline{\text{DA}}$
WF15350S

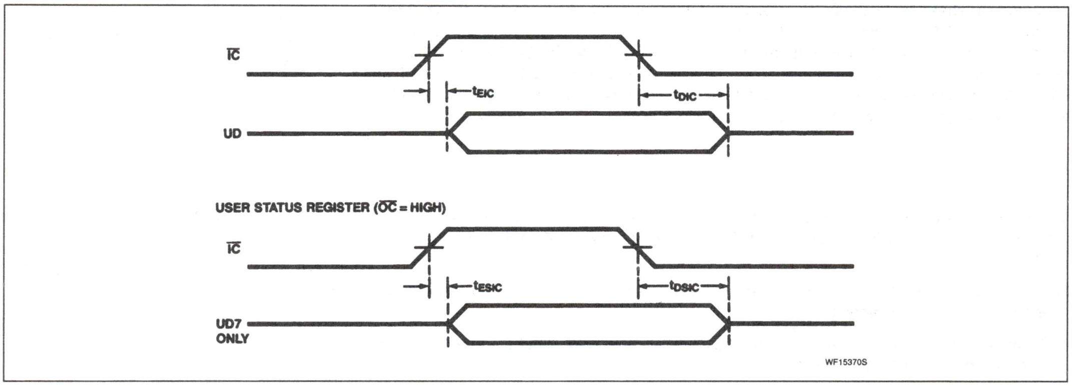
Enable and Disable Timing at the $\overline{\text{DA}}$ Ports


SIGNAL
t_E
t_D
$\overline{\text{DA}}$
WF15360S

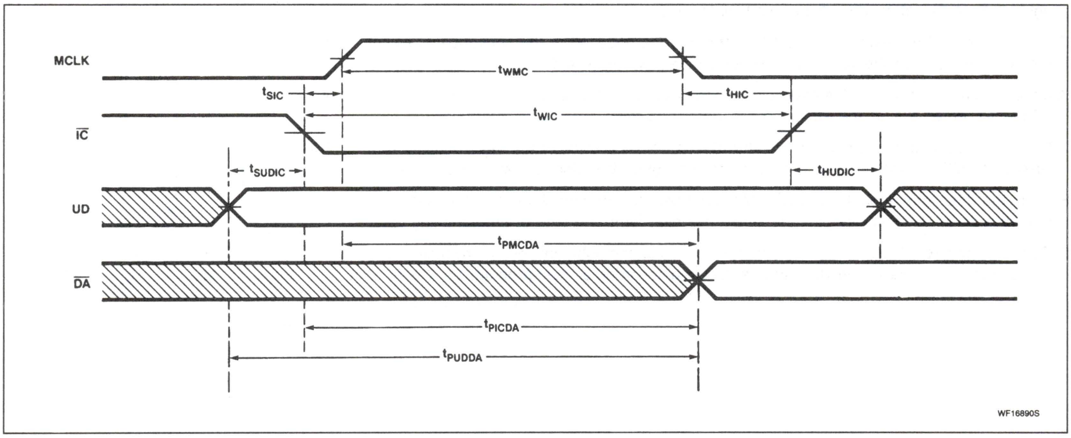
Enable and Disable Signal Summary

SIGNAL	PARAMETERS		CONDITIONS	AFFECT
	Enable	Disable		
$\overline{\text{ME}}$	t_{EME}	t_{DME}	SC = WC = LOW	READ
WC	t_{EWC}	t_{DWC}	SC = $\overline{\text{ME}}$ = LOW	READ
SC	t_{ESC}	t_{DSC}	WC = $\overline{\text{ME}}$ = LOW	READ

Enable and Disable Timing at the UD Ports


$\overline{IC}$
t_{EIC}
t_{DIC}
UD
USER STATUS REGISTER ($\overline{OC}$ = HIGH)
$\overline{IC}$
t_{ESIC}
t_{DSIC}
UD7
ONLY
WF15370S

8X470 Timing-Propagation Delays, Pulse Widths and Hold Times


MCLK
t_{WMC}
t_{SIC}
t_{HIC}
t_{WIC}
$\overline{IC}$
t_{SUDIC}
t_{HUDIC}
UD
t_{PMCDA}
$\overline{DA}$
t_{PICDA}
t_{PUDDA}
WF16890S

8X470 Fuse Coding Sheet

This coding sheet is intended to assist in determining the fuse coding of an 8X470 Programmable I/O Port for a particular application. Use of this sheet requires some prior knowledge of the 8X470 fuse-programmable functions. Refer to the 8X470 Data Sheet for more information. To produce a fuse code, simply follow the steps below. Some of the information will start out in binary form, and must later be converted to hexadecimal. The chart at the right is provided to simplify the conversion process.

Binary to Hex Conversion

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

1. For conventional (DA bus) port selection, choose a Controller Data Address in the range of 0 to FF hexadecimal. Write the address in the space to the right. If Fast Select will be used, enter 00.

   WC - ___

2. Pick the Controller Status Locations. Write a 1 in those positions where status should NOT appear and 0 in the other positions.

   7 6 5 4 3 2 1 0
   ME - _ _ _ _ _ _ _ _

3. Select any bit positions that are to be programmed as input only. Write a 1 in any positions that should be input only, 0 in the other positions.

   7 6 5 4 3 2 1 0
   SC - _ _ _ _ _ _ _ _

4. For conventional (DA bus) port selection, choose the Controller Status Address in the range of F0 to FF hexadecimal. Write the bottom digit (0 to F) of the address in the space to the right. If Fast Select mode will be used, enter a 0.

   IC -

5. If the 8X470 will be used in the Fast Select mode (as opposed to the conventional select mode), enter a 1 in the space to the right, otherwise enter a 0.

   IC -

6. If the 8X470 will be used in the directional mode (Mode B) rather than the bidirectional mode (Mode A), enter a 1 in the space to the right, otherwise enter a 0.

   IC -

7. If it is desired that data be written to the 8X470 from the User Bus asynchronously with respect to MCLK, enter a 1 in the space to the right, otherwise a 0 should be entered.

   IC -

8. Copy the data from lines 4 through 7 into the boxes to the right. As noted, the first bit is always a zero. The four bits plus the single hex digit form one byte of programming data.

   IC - 0 ☐ ☐ ☐ ☐

9. Copy the data from lines 1 to 3 and 8 into the spaces on the right, converting any binary data to hexadecimal. This is the data that is required to order or field-program an 8X470.

   Fuse Code:
   WC ME SC IC
   ___ - ___ - ___ - _

MCCAP 8X300/8X305
Cross Assembler Program

Product Specification (Brief)

Microprocessor Products

DESCRIPTION

The Microcontroller Cross Assembler Program (MCCAP) has been developed to support the Signetics 8X300/8X305 Microcontroller. MCCAP provides many powerful features including macros, automatic subroutine handling, conditional assembly and extended instructions. These features significantly reduce the time required to compose and assemble Microcontroller programs. When combined with standard assembler features such as mnemonic op-codes and address labels, these extended features make MCCAP a powerful programming tool.

As input, MCCAP accepts source code written according to the rules presented in the manual. After assembling the source input, MCCAP produces an assembly listing and machine-readable object module.

MCCAP is written in ANSI standard FORTRAN IV and is available as a fully supported product from Signetics for use on a user's in-house system.

MCCAP is available in several forms. The first is a 9-track magnetic tape containing FORTRAN IV source code for the crossassembler program. For compatibility with various computer systems, the tape is available in various combinations of density and data encoding. To order, use the part numbers in the Ordering Information Table.

The second version of MCCAP runs on an Intel Intellec™ Microcomputer Development System with 64K memory under the control of the ISIS II operating system.

This version (part number 8X300AS2SS) is composed of a two-pass crossassembler program and a PROM formatter overlay. Both programs are written entirely in Intel 8080 Assembly language, and are assembled on the Intel 8080/8085 Macro Assembler version 4.0, linked and located to execute in overlay.

It assembles both 8X300 and 8X305 programs. Also needed is at least one single or double density disk drive with the PROM formatter overlay always residing in drive zero.

The 8X300AS2 software is contained on three diskettes. Disks 1 and 2 contain the Single Density version and disk 3 contains the Double Density version. Both versions will be shipped when ordered under this part number.

The third version of MCCAP is an MS-DOS compatible implementation. This is supplied pre-compiled for immediate use on 5.25″ diskettes. The part number is 8X300AS3SS.

ORDERING INFORMATION

NUMBER	DENSITY	ENCODING
8X300 AS1-1SS	800 BPI	ASCII
8X300 AS1-2SS	800 BPI	EBCDIC
8X300 AS1-3SS	1600 BPI	ASCII
8X300 AS1-4SS	1600 BPI	EBCDIC

Intel Intellec is a trademark of Intel Corporation.

Signetics

FORTRESS 8X401
Cross Assembler Program

Microprocessor Products

DESCRIPTION

The 8X401 assembler (FORTRESS) translates symbolic statements into object code that can be executed by the 8X401 microcontroller. The assembler consists of two passes which build a symbol table, issue helpful error messages, produce a detailed program listing, and output a machine-readable object module.

The 8X401 assembler is written in PASCAL and runs under MS-DOS. It requires a minimum of 128K bytes of memory. If the system does not have a fixed disk, a minimum of a 360K-byte diskette drive is recommended.

FEATURES

- **Free format source code**
- **Predefined symbols for registers, condition codes, length fields, and rotate fields**
- **ASCII character set**
- **Forward referencing**
- **Bit or byte manipulation**
- **User-defined macros**
- **Symbolic references for all operands**
- **Comments for self-documenting code**
- **Cross reference of the symbol table**
- **Support of extended instruction code allowing instructions up to 40-bits wide**
- **Versatile object file format specification**

FORTRESS is supplied on a 5.25″ floppy diskette and may be ordered by the part number 8X400AS1SS.

8X300 KT1SK
8X305 Prototyping System

SYSTEM DESCRIPTION

The 8X305 Prototyping System is a powerful design support tool that aids the engineer in the evaluation, design, and prototyping of systems based on the Signetics 8X305 Microcontroller and its family of support devices. Its advanced features permit the development of both 8X305 firmware and application circuitry. The prototyping systems capabilities are adequate to serve as a complete development system for simple systems and provide a low-cost tool for evaluating portions of more complex designs.

ARCHITECTURAL OVERVIEW

The 8X305 Prototyping System consists of a single printed circuit board that includes an 8X305 Microcontroller and various 8X305 Family peripheral devices. The 8X305's microprogram resides in Writeable Control Storage (WCS). A Monitor Processor controls operation of the 8X305 by loading the WCS, activating the Run/Step logic, and directly placing instructions onto the 8X305's Instruction bus. The Monitor Processor also controls the User Interface, which is through a standard RS-232 connector. The remainder of the board is occupied by power connections and a large wire-wrap area for prototyping of user-developed circuits. A complete discussion of the operation and interrelationship of these functions is contained in later chapters.

User Interface

The User Interface of the 8X305 Prototyping System is accomplished through a standard RS-232 connector. Data rates from 110 to 19,200 baud are switch selected by the user. The monitor program contained in the system controls all user communication, which is accomplished interactively through a straightforward and user-friendly syntax. While the operation of the system requires only a low-cost "dumb" terminal, it can be connected to a host computer to support more advanced developments. Commands are included to support up and down loading of programs in such applications.

Monitor Processor and Run/Step Logic

Operation of the system is controlled by the Monitor Processor, which is implemented using an 8035 Microprocessor. The Monitor Processor is responsible for the following functions:

- User interaction
- Loading Writeable Control Storage
- Loading and reading 8X305 registers and I/O devices
- Control of Run/Step logic

Programming for the Monitor Processor is supplied by Signetics and is contained in an EPROM.

Writeable Control Storage

8X305 Microcontroller programs are executed from a Writeable Control Storage (WCS) that is contained on the board. The prototyping system is supplied with 256 words of instruction memory. An expansion module is available to support address space requirements of up to 4096 words. The WCS is sufficiently fast to permit full speed operation of the 8X305.

WCS words are 25 bits wide to support advanced microprogramming requirements. Sixteen of these bits contain actual 8X305 instructions. Eight of the remaining bits are used to support "Extended Microcode" designs as described in the 8X305 Users Manual. The 25th bit, transparent to the user, is set by the Monitor Processor to control breakpoints.

Since the three-bus architecture of the 8X305 does not permit the Microcontroller to modify its own program memory, the WCS is loaded by the Monitor Processor.

8X305 Microcontroller and Peripherals

An 8X305 Microcontroller and various 8X300 Family peripheral devices are included in the prototyping system.

The Instruction and Program Address busses of the 8X305 are connected to WCS as well as an 8X310 Interrupt Control Coprocessor (ICC). The interrupt and status pins of the 8X310 are available to the user for use in prototyping real-time or other interrupt driven systems.

The 8X305's IV bus is connected to the following 8X300 Family peripheral devices:

(1) 8X320 Bus Interface Register Array
(1) 8X350 256 Byte Bipolar RAM
(1) 8X360 Memory Address Director
(3) 8X372 Addressable 8-bit Bidirectional I/O Ports

IV bus data and control signal connections are also available to the user to permit attachment of other devices or user developed logic. User interface connections to the 8X320, 8X360, and 8X372's are available adjacent to the wire-wrap area to permit prototyping of various 8X305 based designs.

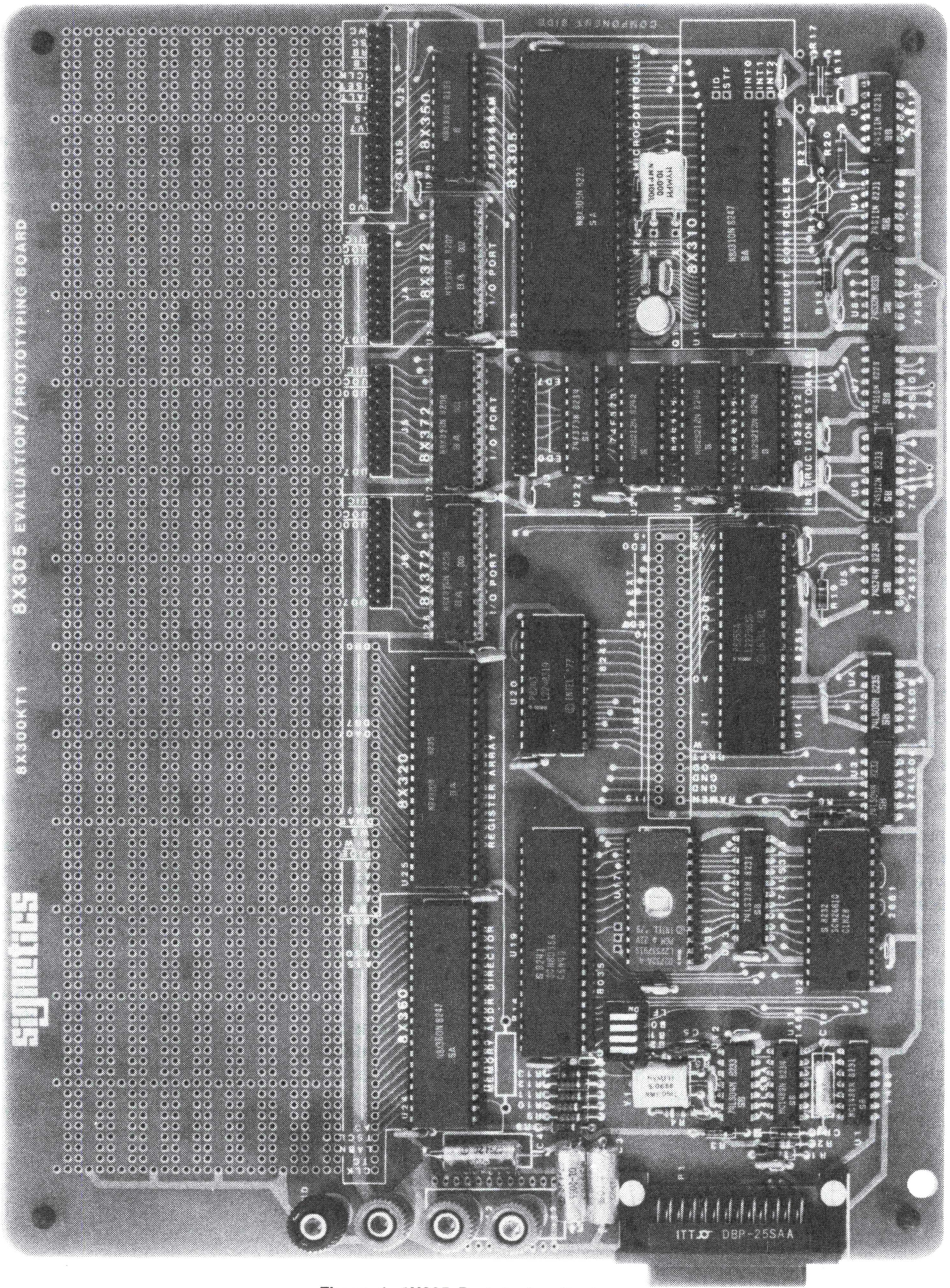

8X305 EVALUATION/PROTOTYPING BOARD
8X300KT1
signetics
COMPONENT SIDE
8X305
MICROCONTROLLER
8X310
INTERRUPT CONTROLLER
INSTRUCTION STORAGE
8X350
8X372
I/O PORT
8X320
REGISTER ARRAY
8X360
ITT DBP-25SAA

Figure 1. 8X305 Prototyping System

Signetics

8X305 Development System

American Automation EZ-PRO

Microprocessor Products

HARDWARE FEATURES

- Real time in-circuit emulation to 10MHz without wait states.
- 8K words of 35ns RAM, 16 or 32-bit words.
- Memory mapping in 1K word increments.
- Compatible with 8X310 Interrupt Control Chip.
- Trace capability includes 128 cycles of address, IV bus, RB, LB, WC, SC, and 3 selectable points in target system. Both input and output phases are traced in each cycle.
- Downloading and Uploading capability provided.
- I/O Port Programmer for 8X372, 8X376, 8X382.

SOFTWARE FEATURES

- Relocating macroassembler compatible with Signetics MCCAP.
- Linking editor permits linking separately-assembled modules to form one load module.
- Debugger program controls single stepping, stopping on a specified address, printing trace information, disassembling, program patching, changing register contents and memory management. Symbolic debug capability is provided.
- Command file capability provided in the Assembler, Linking Editor, and Debugger program.
- PROM formatting program available. Slices 16 and 32-bit words into 4 or 8-bit groups.

EZ-PRO SYSTEM

Devices unique to the 8X300/305 include the AA-572-8X35 In-Circuit Emulator, the AA-572-8X35-M Emulator Extension and the AA-574-8X37 I/O Port Programmer. The Emulator Extension provides an extra sixteen bits of word length over the basic sixteen bits required for the 8X300/305 processor and may or may not be required in a particular development. Note that the emulator consists of three printed circuit board assemblies, while the extension, has two.

The Address Control assembly incorporates trace memory and logic for memory mapping, stopping, and single stepping as well as circuitry required to communicate with the User Interface and Master Processor. Each Emulator Memory assembly is equipped with 8K 16-bit words of 35ns memory as well as interface circuitry. This memory is loaded and unloaded under control of the Master Processor and is accessed by addresses generated by the 8X300/305.

The 8X300/305 User Interface has the processor mounted on it along with cable termination networks, cable drivers, some logic and test points. Test points are provided for the three points which may be traced in the target system, connection to the 8X310 Interrupt Control Chip and oscilloscope sync.

The Extension User Interface is equipped with six DIP sockets and cables which permit connection into PROM sockets located in the target system. Four of the sockets are 18 pin (for 4-bit wide PROMs) and two are 24 pin (for 8-bit wide PROMs). Either set of sockets and cables may be used. Pin outs are compatible with Signetics 82S137, 185, 181, and 191 PROM types.

PROM programming is supported in two ways. Both require that PMFORM, the PROM formatting program, be utilized. After PMFORM has created files consisting of either 4-bit or 8-bit wide slices of the object program words, these files may be directed to a DATA I/O (or equivalent) PROM programmer connected to the RS232 printer port. Alternatively, with 8-bit wide slices, the AA-574-27XX EPROM Programmer may be utilized to write the files into 2716 or 2732 EPROMs. With appropriate off-line equipment, information in the EPROMs may be transferred into bipolar PROMs.

The I/O Port Programmer consists of a printed board assembly, an adaptor that fits into the ZIF socket located on the front of the AA-570 Basic Development Unit and a program. After checking to see that the 8X372/376,382 is properly oriented in the ZIF socket and fuses are unblown, the program permits the desired address to be programmed into the I/O port. Complete checking is then accomplished including validation of the address and transfer of information in both directions through the port.

EZ-PRO SYSTEM ELEMENTS FOR 8X300/305 SUPPORT

Model	Description
AA-570-200	Basic Development Unit
AA-59X	Dual Disk Unit
AA-572-8X35	In-Circuit Emulator for 8X300/305
AA-572-8X35-M	Emulator Extension
AA-562	Printer, RS232 Interface
AA-563	Video Terminal with Screen Editor
AA-553	PMFORM, PROM Formatter Program
A-574-8X37	I/O Port Programmer
AA-574-27XX	EPROM Programmer for 2716 & 2732

Note that all required programs except PMFORM are supplied at no extra cost.

I/O Port Programmer

303A-8X PROGRAMMING TEST ADAPTER

The 303A-8X Programming Test Adapter is designed to program address fuses and activate protect fuses for Signetics' I/O Ports 8X372, 8X374, 8X376, and 8X382. Error messages are displayed if the programmed part is defective or if ambiguous addresses occur during the programming procedure. The test adapter operates in conjunction with the Data I/O Logic PAK 303A-V01 and various models of the Data I/O (System 19, 29A, and 100A). The Programming Test Adapter is quick and easy to use and most Signetics' Franchised Distributors provide on-site programming capabilities for customer parts.

Signetics

Section 4
Package Outlines

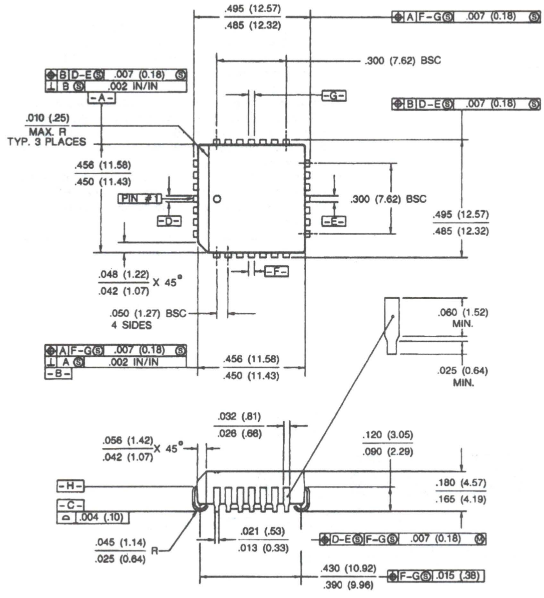
Microprocessor Products

INDEX

A Plastic Leaded Chip Carrier 4-3
D Plastic Small Outline 4-6
F Hermetic CERDIP 4-8
I Hermetic SDIP 4-9
N Plastic DIP 4-13
P Grid Array 4-18

Package Outlines

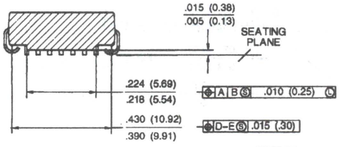
28-PIN PLASTIC LEADED CHIP CARRIER


.495 (12.57)
.485 (12.32)
A F-G S .007 (0.18) S
.300 (7.62) BSC
B D-E S .007 (0.18) S
B S .002 IN/IN
-A-
-G-
B D-E S .007 (0.18) S
.010 (.25)
MAX. R
TYP. 3 PLACES
.456 (11.58)
.450 (11.43)
PIN #1
.300 (7.62) BSC
-D-
-E-
.495 (12.57)
.485 (12.32)
.048 (1.22)
.042 (1.07)
X 45°
-F-
.050 (1.27) BSC
4 SIDES
.060 (1.52)
MIN.
A F-G S .007 (0.18) S
A S .002 IN/IN
-B-
.456 (11.58)
.450 (11.43)
.025 (0.64)
MIN.
.032 (.81)
.026 (.66)
.056 (1.42)
.042 (1.07)
X 45°
.120 (3.05)
.090 (2.29)
-H-
-C-
.004 (.10)
.180 (4.57)
.165 (4.19)
.021 (.53)
.013 (0.33)
D-E S F-G S .007 (0.18) M
.045 (1.14)
.025 (0.64)
R
.430 (10.92)
.390 (9.96)
F-G S .015 (.38)

853-0401 84711

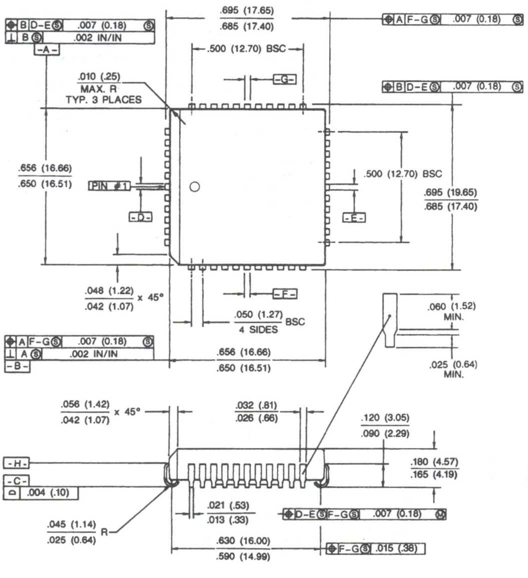
NOTES:

1. Package dimensions conform to JEDEC specification MO-047-AB for plastic leaded chip carrier 28 leads, .050 inch lead spacing, square. (issue A, 10/31/84).
2. Controlling dimensions: inches. Metric are shown in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M-1982.
4. "A" and "B" are reference datums on the molded body at plane "H" and do not include mold flash. Mold flash protrusion shall not exceed .006" (.15mm) on any side.
5. Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
6. Pin numbers continue counterclockwise to pin #28 (top view).
7. Signetics order code for product packaged in a PLCC is the suffix A after the product number.


.015 (0.38)
.005 (0.13)
SEATING
PLANE
.224 (5.69)
.218 (5.54)
A B S .010 (0.25) L
.430 (10.92)
.390 (9.91)
D-E S .015 (.30)

PO00613S

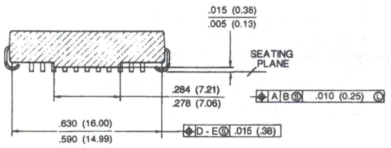
44-PIN PLASTIC LEADED CHIP CARRIER


.695 (17.65)
.685 (17.40)
A F-G S .007 (0.18) S
B D-E S .007 (0.18) S
B S .002 IN/IN
-A-
.500 (12.70) BSC
-G-
B D-E S .007 (0.18) S
.010 (.25)
MAX. R
TYP. 3 PLACES
.656 (16.66)
.650 (16.51)
PIN #1
.500 (12.70) BSC
.695 (19.65)
.685 (17.40)
-D-
-E-
.048 (1.22)
.042 (1.07)
x 45°
-F-
.050 (1.27)
4 SIDES
BSC
.060 (1.52)
MIN.
A F-G S .007 (0.18) S
A S .002 IN/IN
-B-
.656 (16.66)
.650 (16.51)
.025 (0.64)
MIN.
.056 (1.42)
.042 (1.07)
x 45°
.032 (.81)
.026 (.66)
.120 (3.05)
.090 (2.29)
-H-
-C-
.004 (.10)
.180 (4.57)
.165 (4.19)
.021 (.53)
.013 (.33)
D-E S F-G S .007 (0.18) M
.045 (1.14)
.025 (0.64)
R
.630 (16.00)
.590 (14.99)
F-G S .015 (.38)

853-0403 84711

NOTES:

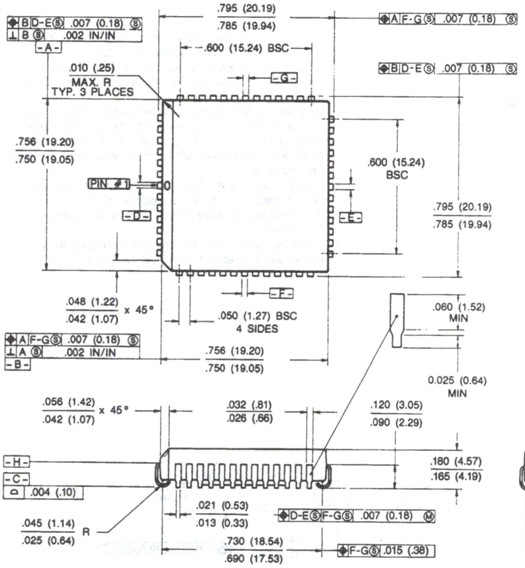
1. Package dimensions conform to JEDEC specification MO-047-AL for plastic leaded chip carrier 44 leads, .050 inch lead spacing, square. (issue A, 10/31/84).
2. Controlling dimensions: inches. Metric are shown in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M-1982.
4. "A" and "B" are reference datums on the molded body at plane "H" and do not include mold flash. Mold flash protrusion shall not exceed .006" (.15mm) on any side.
5. Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
6. Pin numbers continue counterclockwise to pin #44 (top view).
7. Signetics order code for product packaged in a PLCC is the suffix A after the product number.

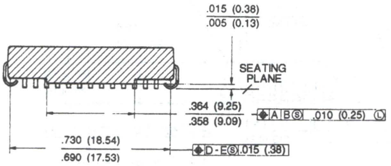

.015 (0.38)
.005 (0.13)
SEATING
PLANE
.284 (7.21)
.278 (7.06)
A B S .010 (0.25) L
.630 (16.00)
.590 (14.99)
D-E S .015 (.38)

PO00603S

Package Outlines

52-PIN PLASTIC LEADED CHIP CARRIER


⌖ B D-E Ⓢ .007 (0.18) Ⓢ
⊥ B Ⓢ .002 IN/IN
-A-
.795 (20.19)
.785 (19.94)
.600 (15.24) BSC
⌖ A F-G Ⓢ .007 (0.18) Ⓢ
⌖ B D-E Ⓢ .007 (0.18) Ⓢ
.010 (.25)
MAX. R
TYP. 3 PLACES
-G-
.756 (19.20)
.750 (19.05)
PIN #1
-D-
.600 (15.24)
BSC
-E-
.795 (20.19)
.785 (19.94)
-F-
.048 (1.22)
.042 (1.07)
x 45°
.050 (1.27) BSC
4 SIDES
.060 (1.52)
MIN
⌖ A F-G Ⓢ .007 (0.18) Ⓢ
⊥ A Ⓢ .002 IN/IN
-B-
.756 (19.20)
.750 (19.05)
0.025 (0.64)
MIN
.056 (1.42)
.042 (1.07)
x 45°
.032 (.81)
.026 (.66)
.120 (3.05)
.090 (2.29)
-H-
-C-
⏥ .004 (.10)
.180 (4.57)
.165 (4.19)
.021 (0.53)
.013 (0.33)
⌖ D-E Ⓢ F-G Ⓢ .007 (0.18) Ⓜ
.045 (1.14)
.025 (0.64)
R
.730 (18.54)
.690 (17.53)
⌖ F-G Ⓢ .015 (.38)


.015 (0.38)
.005 (0.13)
SEATING
PLANE
.364 (9.25)
.358 (9.09)
⌖ A B Ⓢ .010 (0.25) Ⓛ
.730 (18.54)
.690 (17.53)
⌖ D-E Ⓢ .015 (.38)

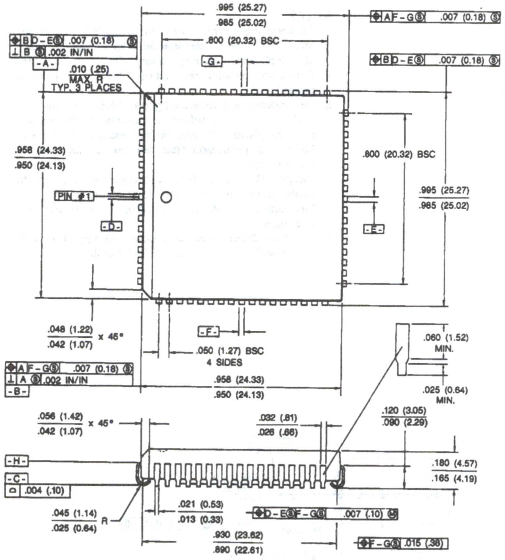
NOTES:

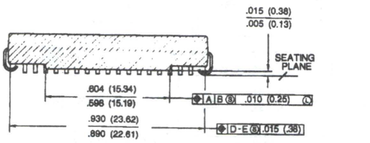
1. Package dimensions conform to JEDEC specification MO-047-AD for plastic leaded chip carrier 52 leads, .050 inch lead spacing, square. (issue A, 10/31/84).
2. Controlling dimensions: inches. Metric are shown in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M-1982.
4. "A" and "B" are reference datums on the molded body at plane "H" and do not include mold flash. Mold flash protrusion shall not exceed .006" (.15mm) on any side.
5. Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
6. Pin numbers continue counterclockwise to pin #52 (top view).
7. Signetics order code for product packaged in a PLCC is the suffix A after the product number.

853-0397 84711

PO00644S

68-PIN PLASTIC LEADED CHIP CARRIER


.995 (25.27)
.985 (25.02)
⌖ A F-G Ⓢ .007 (0.18) Ⓢ
⌖ B D-E Ⓢ .007 (0.18) Ⓢ
⊥ B Ⓢ .002 IN/IN
-A-
.800 (20.32) BSC
-G-
⌖ B D-E Ⓢ .007 (0.18) Ⓢ
.010 (.25)
MAX. R
TYP. 3 PLACES
.958 (24.33)
.950 (24.13)
.800 (20.32) BSC
PIN #1
.995 (25.27)
.985 (25.02)
-D-
-E-
.048 (1.22)
.042 (1.07)
x 45°
-F-
.050 (1.27) BSC
4 SIDES
.060 (1.52)
MIN.
⌖ A F-G Ⓢ .007 (0.18) Ⓢ
⊥ A Ⓢ .002 IN/IN
-B-
.958 (24.33)
.950 (24.13)
.025 (0.64)
MIN.
.056 (1.42)
.042 (1.07)
x 45°
.032 (.81)
.026 (.66)
.120 (3.05)
.090 (2.29)
-H-
-C-
⏥ .004 (.10)
.180 (4.57)
.165 (4.19)
.045 (1.14)
.025 (0.64)
R
.021 (0.53)
.013 (0.33)
⌖ D-E Ⓢ F-G Ⓢ .007 (.10) Ⓜ
.930 (23.62)
.890 (22.61)
⌖ F-G Ⓢ .015 (.38)


.015 (0.38)
.005 (0.13)
SEATING
PLANE
.604 (15.34)
.598 (15.19)
⌖ A B Ⓢ .010 (0.25) Ⓛ
.930 (23.62)
.890 (22.61)
⌖ D-E Ⓢ .015 (.38)

NOTES:

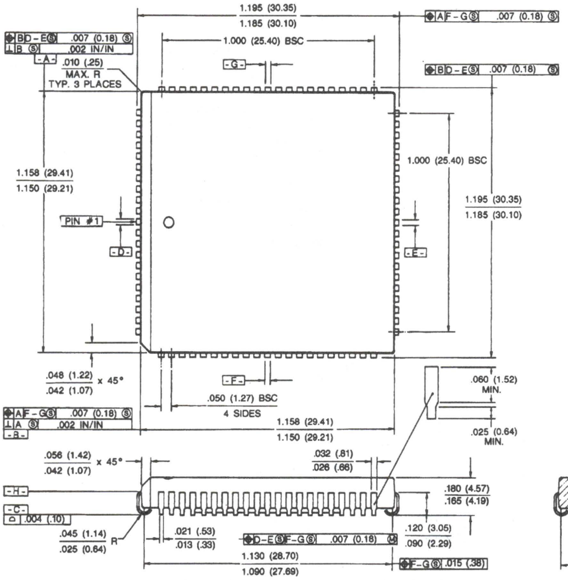
1. Package dimensions conform to JEDEC specification MO-047-AE for plastic leaded chip carrier 68 leads, .050 inch lead spacing, square. (issue A, 10/31/84).
2. Controlling dimensions: inches. Metric are shown in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M-1982.
4. "A" and "B" are reference datums on the molded body at plane "H" and do not include mold flash. Mold flash protrusion shall not exceed .006" (.15mm) on any side.
5. Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
6. Pin numbers continue counterclockwise to pin #68 (top view).
7. Signetics order code for product packaged in a PLCC is the suffix A after the product number.

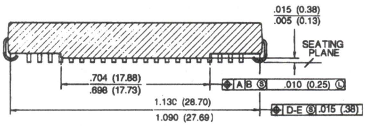
853-0398 84711

PO00623S

Package Outlines

84-PIN PLASTIC LEADED CHIP CARRIER


1.195 (30.35)
1.185 (30.10)
AF-G Ⓢ .007 (0.18) Ⓢ
BD-E Ⓢ .007 (0.18) Ⓢ
⊥ B Ⓢ .002 IN/IN
-A-
.010 (.25)
MAX. R
TYP. 3 PLACES
1.000 (25.40) BSC
-G-
BD-E Ⓢ .007 (0.18) Ⓢ
1.000 (25.40) BSC
1.158 (29.41)
1.150 (29.21)
1.195 (30.35)
1.185 (30.10)
PIN #1
-D-
-E-
.048 (1.22)
.042 (1.07)
x 45°
-F-
.050 (1.27) BSC
4 SIDES
.060 (1.52)
MIN.
AF-G Ⓢ .007 (0.18) Ⓢ
⊥ A Ⓢ .002 IN/IN
-B-
1.158 (29.41)
1.150 (29.21)
.025 (0.64)
MIN.
.056 (1.42)
.042 (1.07)
x 45°
.032 (.81)
.026 (.66)
-H-
.180 (4.57)
.165 (4.19)
-C-
.004 (.10)
.045 (1.14)
.025 (0.64)
R
.021 (.53)
.013 (.33)
D-E Ⓢ F-G Ⓢ .007 (0.18) Ⓜ
.120 (3.05)
.090 (2.29)
1.130 (28.70)
1.090 (27.69)
F-G Ⓢ .015 (.38)


.015 (0.38)
.005 (0.13)
SEATING PLANE
.704 (17.88)
.698 (17.73)
A B Ⓢ .010 (0.25) Ⓛ
1.13C (28.70)
1.090 (27.69)
D-E Ⓢ .015 (.38)

NOTES:

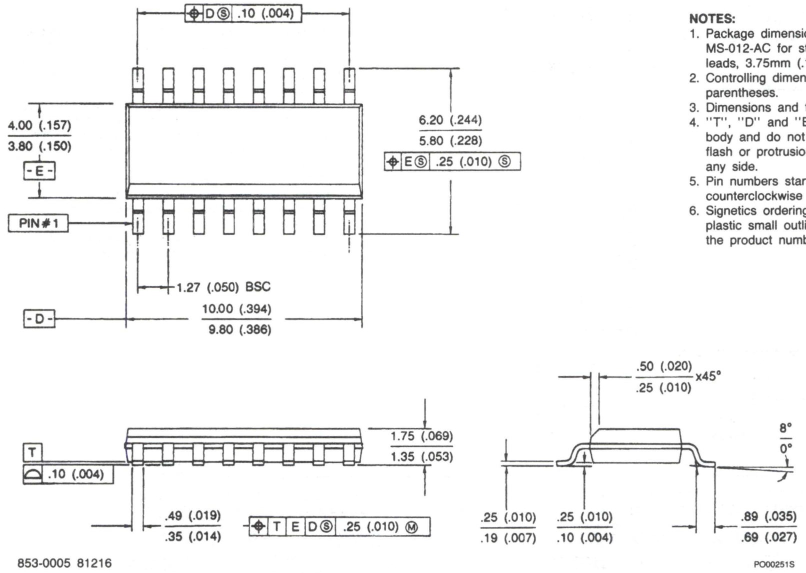
1. Package dimensions conform to JEDEC specification MO-047-AF for plastic leaded chip carrier 84 leads, .050 inch lead spacing, square. (issue A, 10/31/84).
2. Controlling dimensions: inches. Metric are shown in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M-1982.
4. "A" and "B" are reference datums on the molded body at plane "H" and do not include mold flash. Mold flash protrusion shall not exceed .006" (.15mm) on any side.
5. Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
6. Pin numbers continue counterclockwise to pin #84 (top view).
7. Signetics order code for product packaged in a PLCC is the suffix A after the product number.

853-0399 84711

PO00593S

Package Outlines

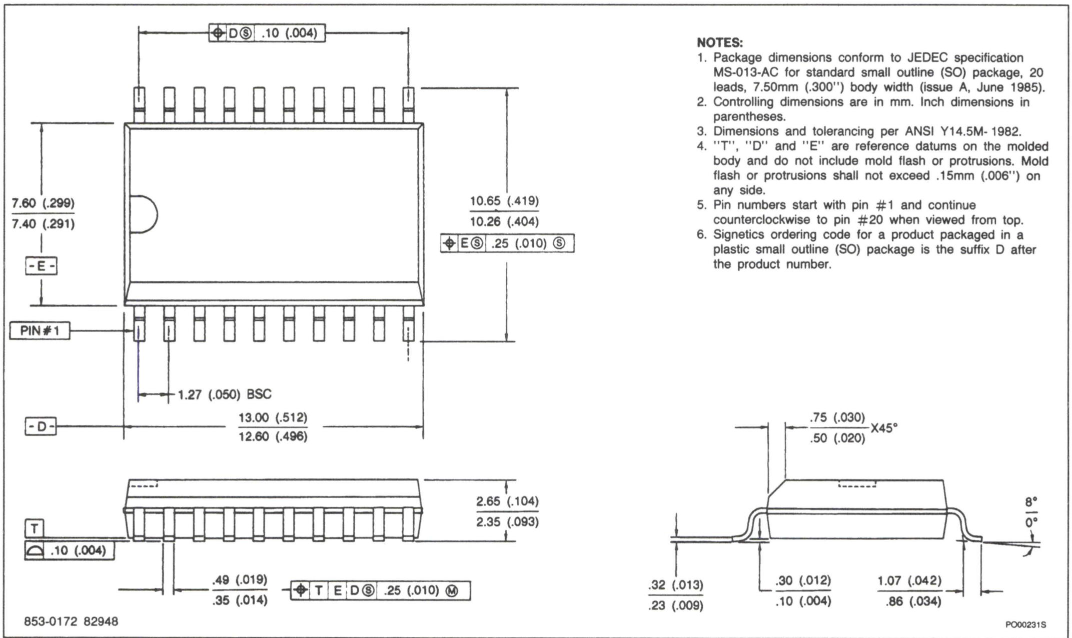
16-PIN PLASTIC SO


⊕ D Ⓢ .10 (.004)
4.00 (.157)
3.80 (.150)
-E-
6.20 (.244)
5.80 (.228)
⊕ E Ⓢ .25 (.010) Ⓢ
PIN#1
1.27 (.050) BSC
-D-
10.00 (.394)
9.80 (.386)
.50 (.020)
.25 (.010)
x45°
T
1.75 (.069)
1.35 (.053)
8°
0°
.10 (.004)
.49 (.019)
.35 (.014)
⊕ T E D Ⓢ .25 (.010) Ⓜ
.25 (.010)
.19 (.007)
.25 (.010)
.10 (.004)
.89 (.035)
.69 (.027)
853-0005 81216
PO00251S

NOTES:

1. Package dimensions conform to JEDEC specification MS-012-AC for standard small outline (SO) package, 16 leads, 3.75mm (.150") body width (issue A, June 1985).
2. Controlling dimensions are in mm. Inch dimensions in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M-1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .15mm (.006") on any side.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #16 when viewed from top.
6. Signetics ordering code for a product packaged in a plastic small outline (SO) package is the suffix D after the product number.

20-PIN PLASTIC SOL

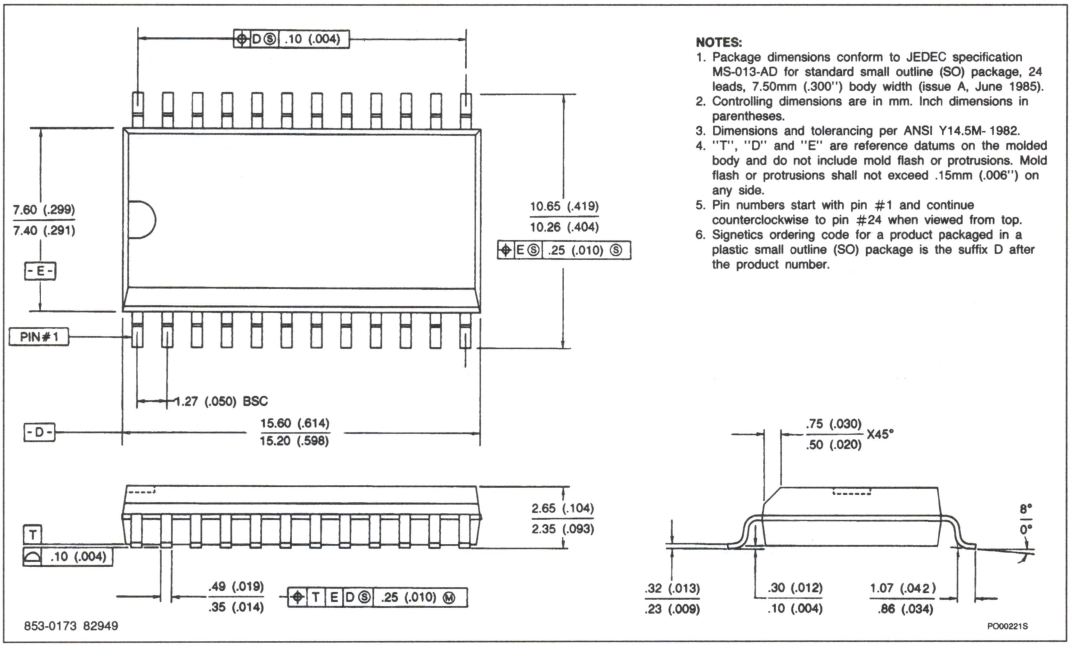

⊕ D Ⓢ .10 (.004)
7.60 (.299)
7.40 (.291)
-E-
10.65 (.419)
10.26 (.404)
⊕ E Ⓢ .25 (.010) Ⓢ
PIN#1
1.27 (.050) BSC
-D-
13.00 (.512)
12.60 (.496)
.75 (.030)
.50 (.020)
X45°
T
2.65 (.104)
2.35 (.093)
8°
0°
.10 (.004)
.49 (.019)
.35 (.014)
⊕ T E D Ⓢ .25 (.010) Ⓜ
.32 (.013)
.23 (.009)
.30 (.012)
.10 (.004)
1.07 (.042)
.86 (.034)
853-0172 82948
PO00231S

NOTES:

1. Package dimensions conform to JEDEC specification MS-013-AC for standard small outline (SO) package, 20 leads, 7.50mm (.300") body width (issue A, June 1985).
2. Controlling dimensions are in mm. Inch dimensions in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M-1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .15mm (.006") on any side.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #20 when viewed from top.
6. Signetics ordering code for a product packaged in a plastic small outline (SO) package is the suffix D after the product number.

Package Outlines

24-PIN PLASTIC SOL


⊕ D Ⓢ .10 (.004)
7.60 (.299)
7.40 (.291)
-E-
PIN#1
10.65 (.419)
10.26 (.404)
⊕ E Ⓢ .25 (.010) Ⓢ
1.27 (.050) BSC
-D-
15.60 (.614)
15.20 (.598)
T
.10 (.004)
2.65 (.104)
2.35 (.093)
.49 (.019)
.35 (.014)
⊕ T E D Ⓢ .25 (.010) Ⓜ
.75 (.030)
.50 (.020)
X45°
8°
0°
.32 (.013)
.23 (.009)
.30 (.012)
.10 (.004)
1.07 (.042)
.86 (.034)
853-0173 82949
PO00221S

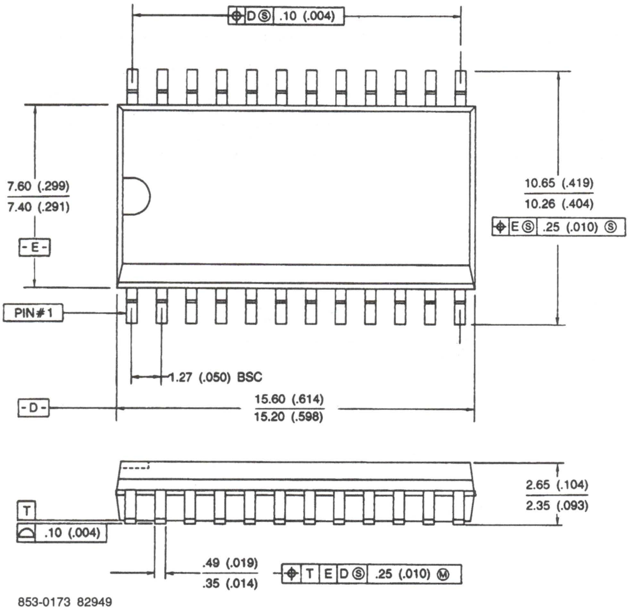
NOTES:

1. Package dimensions conform to JEDEC specification MS-013-AD for standard small outline (SO) package, 24 leads, 7.50mm (.300") body width (issue A, June 1985).
2. Controlling dimensions are in mm. Inch dimensions in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M- 1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .15mm (.006") on any side.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #24 when viewed from top.
6. Signetics ordering code for a product packaged in a plastic small outline (SO) package is the suffix D after the product number.

NOTES

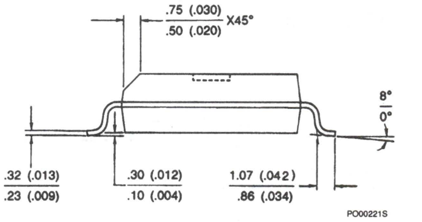
Package Outlines

24-PIN PLASTIC SOL


⌖ D Ⓢ .10 (.004)
7.60 (.299)
7.40 (.291)
- E -
PIN#1
10.65 (.419)
10.26 (.404)
⌖ E Ⓢ .25 (.010) Ⓢ
1.27 (.050) BSC
- D -
15.60 (.614)
15.20 (.598)
T
⌓ .10 (.004)
2.65 (.104)
2.35 (.093)
.49 (.019)
.35 (.014)
⌖ T E D Ⓢ .25 (.010) Ⓜ
853-0173 82949

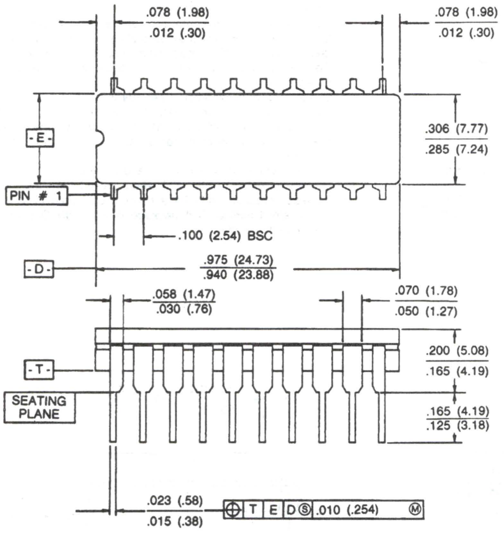
NOTES:

1. Package dimensions conform to JEDEC specification MS-013-AD for standard small outline (SO) package, 24 leads, 7.50mm (.300") body width (issue A, June 1985).
2. Controlling dimensions are in mm. Inch dimensions in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M- 1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .15mm (.006") on any side.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #24 when viewed from top.
6. Signetics ordering code for a product packaged in a plastic small outline (SO) package is the suffix D after the product number.


.75 (.030)
.50 (.020)
X45°
8°
0°
.32 (.013)
.23 (.009)
.30 (.012)
.10 (.004)
1.07 (.042)
.86 (.034)
PO00221S

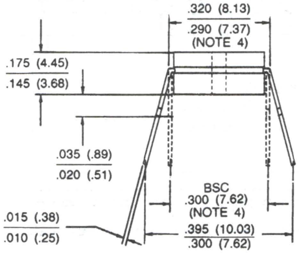
Package Outlines

20-PIN HERMETIC CERDIP


.078 (1.98)
.012 (.30)
.078 (1.98)
.012 (.30)
-E-
.306 (7.77)
.285 (7.24)
PIN # 1
.100 (2.54) BSC
-D-
.975 (24.73)
.940 (23.88)
.058 (1.47)
.030 (.76)
.070 (1.78)
.050 (1.27)
-T-
SEATING PLANE
.200 (5.08)
.165 (4.19)
.165 (4.19)
.125 (3.18)
.023 (.58)
.015 (.38)
⊕ T E D Ⓢ .010 (.254) Ⓜ

NOTES:

1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M - 1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #20 when viewed from the top.

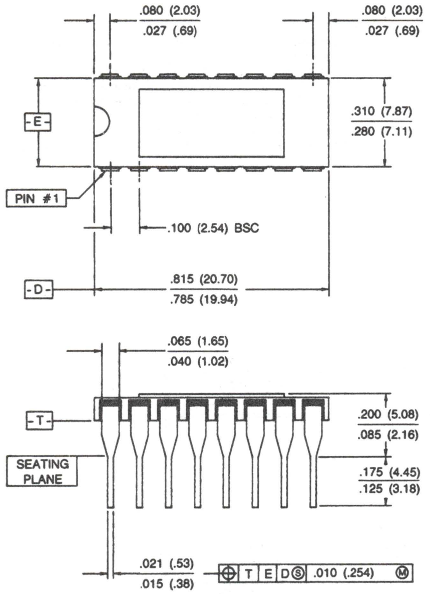

.320 (8.13)
.290 (7.37)
(NOTE 4)
.175 (4.45)
.145 (3.68)
.035 (.89)
.020 (.51)
BSC
.300 (7.62)
(NOTE 4)
.015 (.38)
.010 (.25)
.395 (10.03)
.300 (7.62)

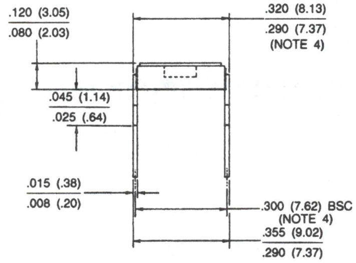
853-0584 81594

PO00430S

Package Outlines

16-PIN HERMETIC SDIP


.080 (2.03)
.027 (.69)
.080 (2.03)
.027 (.69)
-E-
.310 (7.87)
.280 (7.11)
PIN #1
.100 (2.54) BSC
-D-
.815 (20.70)
.785 (19.94)
.065 (1.65)
.040 (1.02)
-T-
SEATING PLANE
.200 (5.08)
.085 (2.16)
.175 (4.45)
.125 (3.18)
.021 (.53)
.015 (.38)
⊕ T E D Ⓢ .010 (.254) Ⓜ

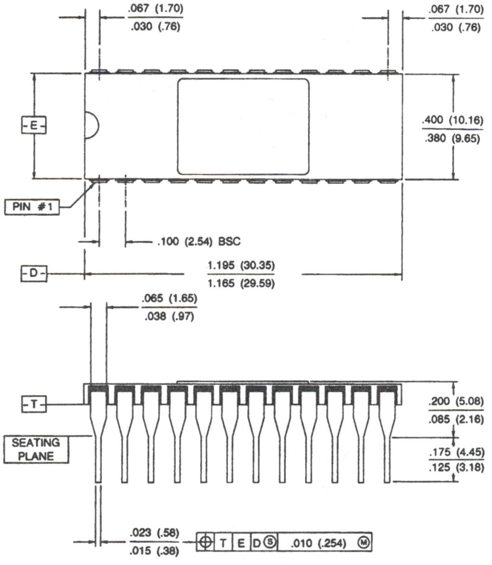

.120 (3.05)
.080 (2.03)
.320 (8.13)
.290 (7.37)
(NOTE 4)
.045 (1.14)
.025 (.64)
.015 (.38)
.008 (.20)
.300 (7.62) BSC
(NOTE 4)
.355 (9.02)
.290 (7.37)

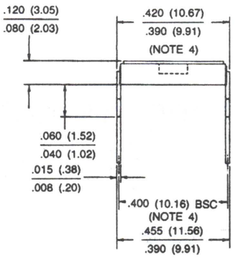
NOTES:
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the body.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #16 when viewed from the top.

853-0109 82276

PO00530S

24-PIN HERMETIC SDIP


.067 (1.70)
.030 (.76)
.067 (1.70)
.030 (.76)
-E-
.400 (10.16)
.380 (9.65)
PIN #1
.100 (2.54) BSC
-D-
1.195 (30.35)
1.165 (29.59)
.065 (1.65)
.038 (.97)
-T-
SEATING PLANE
.200 (5.08)
.085 (2.16)
.175 (4.45)
.125 (3.18)
.023 (.58)
.015 (.38)
⊕ T E D Ⓢ .010 (.254) Ⓜ


.120 (3.05)
.080 (2.03)
.420 (10.67)
.390 (9.91)
(NOTE 4)
.060 (1.52)
.040 (1.02)
.015 (.38)
.008 (.20)
.400 (10.16) BSC
(NOTE 4)
.455 (11.56)
.390 (9.91)

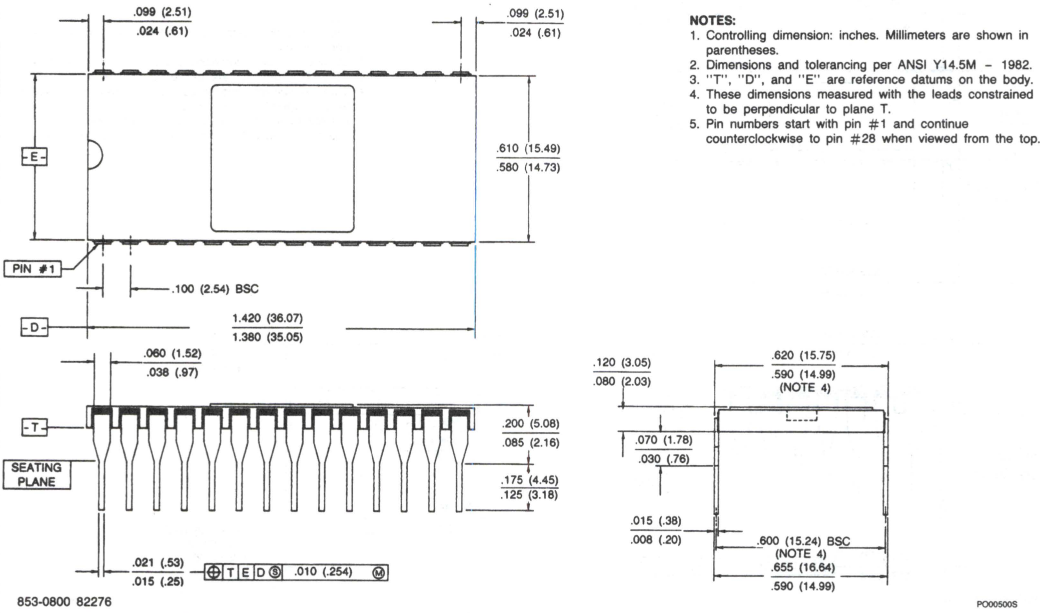
NOTES:
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the body.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #24 when viewed from the top.

853-0110 82276

PO00551S

Package Outlines

28-PIN HERMETIC SDIP

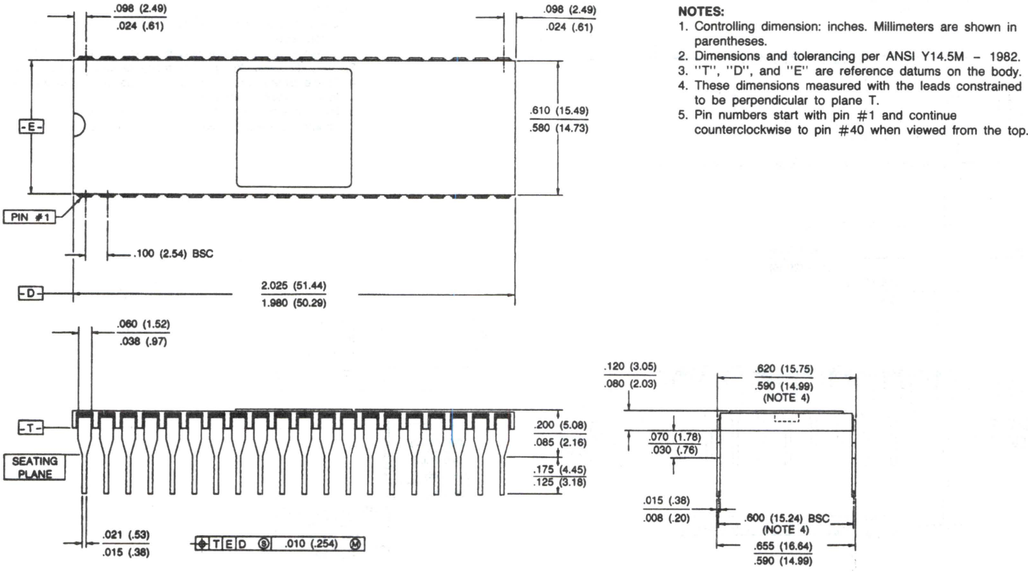

.099 (2.51)
.024 (.61)
.099 (2.51)
.024 (.61)
-E-
.610 (15.49)
.580 (14.73)
PIN #1
.100 (2.54) BSC
-D-
1.420 (36.07)
1.380 (35.05)
.060 (1.52)
.038 (.97)
-T-
SEATING PLANE
.200 (5.08)
.085 (2.16)
.175 (4.45)
.125 (3.18)
.021 (.53)
.015 (.25)
⊕ T E D Ⓢ .010 (.254) Ⓜ
.120 (3.05)
.080 (2.03)
.620 (15.75)
.590 (14.99)
(NOTE 4)
.070 (1.78)
.030 (.76)
.015 (.38)
.008 (.20)
.600 (15.24) BSC
(NOTE 4)
.655 (16.64)
.590 (14.99)

NOTES:
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the body.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #28 when viewed from the top.

853-0800 82276

PO00500S

40-PIN HERMETIC SDIP


.098 (2.49)
.024 (.61)
.098 (2.49)
.024 (.61)
-E-
.610 (15.49)
.580 (14.73)
PIN #1
.100 (2.54) BSC
-D-
2.025 (51.44)
1.980 (50.29)
.060 (1.52)
.038 (.97)
-T-
SEATING PLANE
.200 (5.08)
.085 (2.16)
.175 (4.45)
.125 (3.18)
.021 (.53)
.015 (.38)
⊕ T E D Ⓢ .010 (.254) Ⓜ
.120 (3.05)
.080 (2.03)
.620 (15.75)
.590 (14.99)
(NOTE 4)
.070 (1.78)
.030 (.76)
.015 (.38)
.008 (.20)
.600 (15.24) BSC
(NOTE 4)
.655 (16.64)
.590 (14.99)

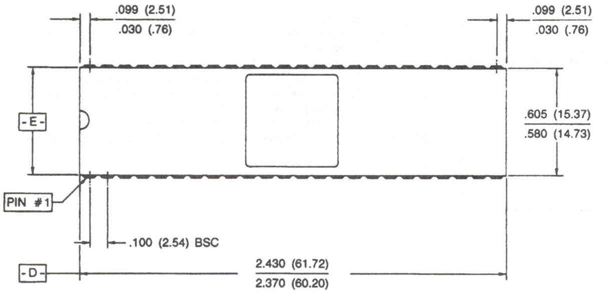
NOTES:
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the body.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #40 when viewed from the top.

853-0801 82276

PO00580S

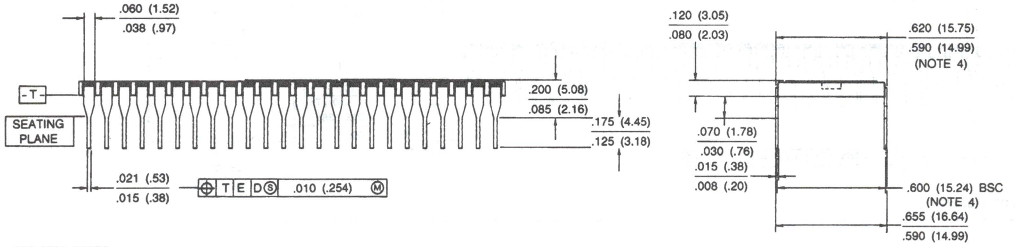
Package Outlines

48-PIN HERMETIC SDIP


.099 (2.51)
.030 (.76)
.099 (2.51)
.030 (.76)
-E-
.605 (15.37)
.580 (14.73)
PIN #1
.100 (2.54) BSC
-D-
2.430 (61.72)
2.370 (60.20)

NOTES:

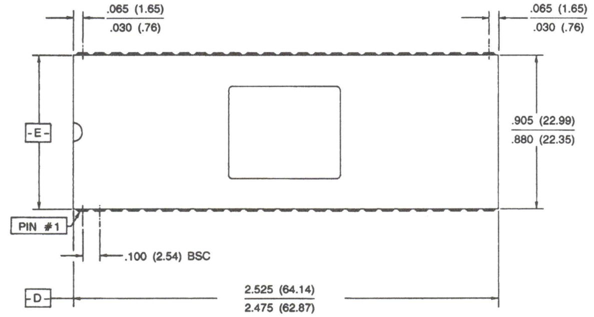
1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the body.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #48 when viewed from the top.


.060 (1.52)
.038 (.97)
-T-
SEATING PLANE
.200 (5.08)
.085 (2.16)
.175 (4.45)
.125 (3.18)
.021 (.53)
.015 (.38)
T E D S .010 (.254) M
.120 (3.05)
.080 (2.03)
.620 (15.75)
.590 (14.99)
(NOTE 4)
.070 (1.78)
.030 (.76)
.015 (.38)
.008 (.20)
.600 (15.24) BSC
(NOTE 4)
.655 (16.64)
.590 (14.99)

853-0802 82276

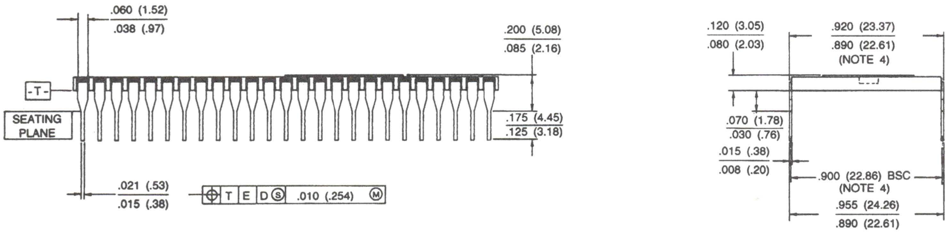
PO00521S

50-PIN HERMETIC SDIP


.065 (1.65)
.030 (.76)
.065 (1.65)
.030 (.76)
-E-
.905 (22.99)
.880 (22.35)
PIN #1
.100 (2.54) BSC
-D-
2.525 (64.14)
2.475 (62.87)

NOTES:

1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the body.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #50 when viewed from the top.

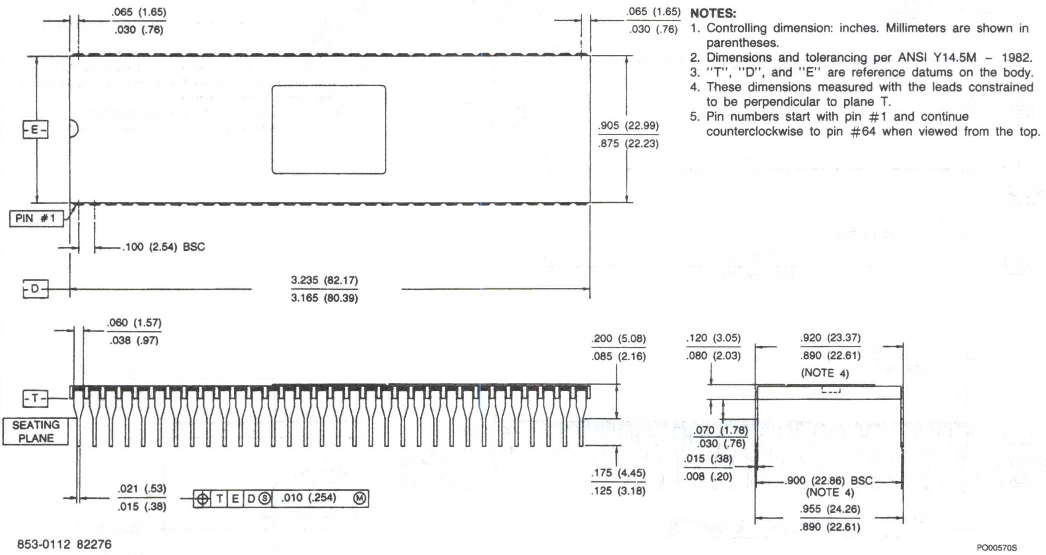

.060 (1.52)
.038 (.97)
.200 (5.08)
.085 (2.16)
-T-
SEATING PLANE
.175 (4.45)
.125 (3.18)
.021 (.53)
.015 (.38)
T E D S .010 (.254) M
.120 (3.05)
.080 (2.03)
.920 (23.37)
.890 (22.61)
(NOTE 4)
.070 (1.78)
.030 (.76)
.015 (.38)
.008 (.20)
.900 (22.86) BSC
(NOTE 4)
.955 (24.26)
.890 (22.61)

853-0807 82276

PO00560S

Package Outlines

64-PIN HERMETIC SDIP

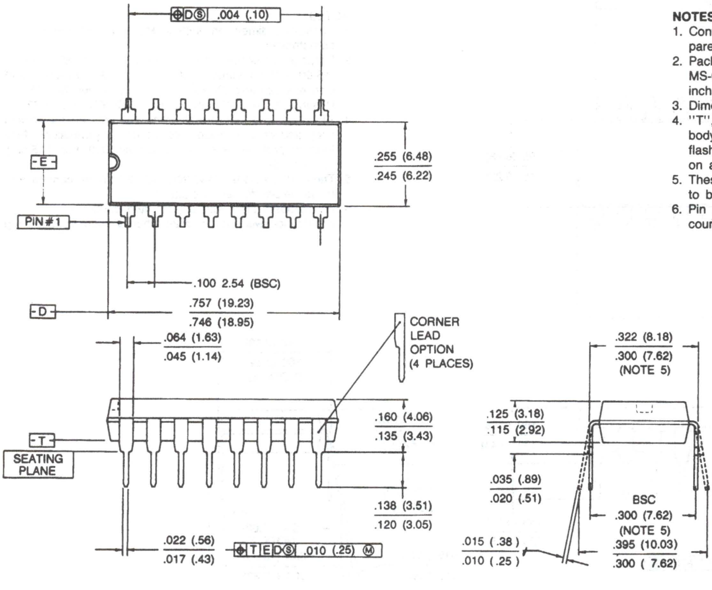

.065 (1.65)
.030 (.76)
.065 (1.65)
.030 (.76)
-E-
.905 (22.99)
.875 (22.23)
PIN #1
.100 (2.54) BSC
-D-
3.235 (82.17)
3.165 (80.39)
.060 (1.57)
.038 (.97)
.200 (5.08)
.085 (2.16)
.120 (3.05)
.080 (2.03)
.920 (23.37)
.890 (22.61)
(NOTE 4)
-T-
SEATING PLANE
.070 (1.78)
.030 (.76)
.015 (.38)
.008 (.20)
.175 (4.45)
.125 (3.18)
.900 (22.86) BSC
(NOTE 4)
.955 (24.26)
.890 (22.61)
.021 (.53)
.015 (.38)
⌖ T E D Ⓢ .010 (.254) Ⓜ
853-0112 82276
PO00570S

NOTES:

1. Controlling dimension: inches. Millimeters are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the body.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #64 when viewed from the top.

Package Outlines

16-PIN PLASTIC DIP

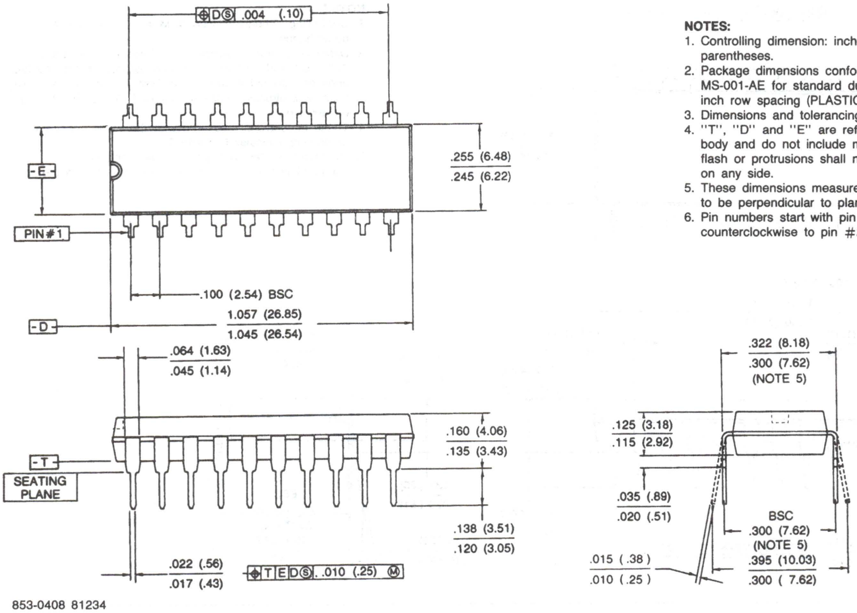


D S .004 (.10)
-E-
PIN #1
.255 (6.48)
.245 (6.22)
.100 2.54 (BSC)
-D-
.757 (19.23)
.746 (18.95)
.064 (1.63)
.045 (1.14)
CORNER LEAD OPTION (4 PLACES)
-T-
SEATING PLANE
.160 (4.06)
.135 (3.43)
.138 (3.51)
.120 (3.05)
.022 (.56)
.017 (.43)
T E D S .010 (.25) M
.322 (8.18)
.300 (7.62)
(NOTE 5)
.125 (3.18)
.115 (2.92)
.035 (.89)
.020 (.51)
BSC
.300 (7.62)
(NOTE 5)
.015 (.38)
.010 (.25)
.395 (10.03)
.300 (7.62)
853-0406 81232
PO00280S

NOTES:

1. Controlling dimension: inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC specification MS-001-AA for standard dual in-line (DIP) package .300 inch row spacing (PLASTIC) 16 leads (issue B. 7/85)
3. Dimensions and tolerancing per ANSI Y14. 5M-1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with pin #1 and continue counterclockwise to pin #16 when viewed from the top.

20-PIN PLASTIC DIP



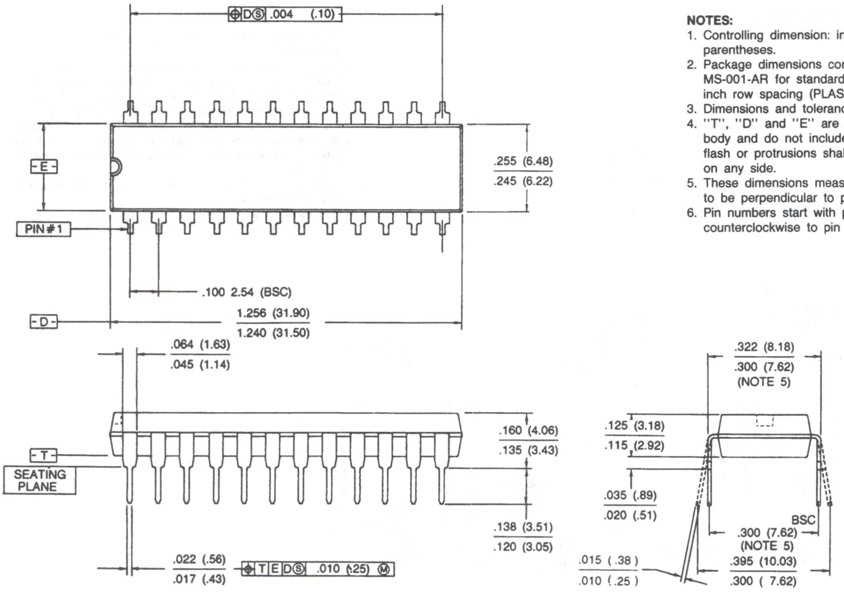
D S .004 (.10)
-E-
PIN #1
.255 (6.48)
.245 (6.22)
.100 (2.54) BSC
-D-
1.057 (26.85)
1.045 (26.54)
.064 (1.63)
.045 (1.14)
-T-
SEATING PLANE
.160 (4.06)
.135 (3.43)
.138 (3.51)
.120 (3.05)
.022 (.56)
.017 (.43)
T E D S .010 (.25) M
.322 (8.18)
.300 (7.62)
(NOTE 5)
.125 (3.18)
.115 (2.92)
.035 (.89)
.020 (.51)
BSC
.300 (7.62)
(NOTE 5)
.015 (.38)
.010 (.25)
.395 (10.03)
.300 (7.62)
853-0408 81234
PO00290S

NOTES:

1. Controlling dimension: inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC specification MS-001-AE for standard dual in-line (DIP) package .300 inch row spacing (PLASTIC) 20 leads (issue B. 7/85)
3. Dimensions and tolerancing per ANSI Y14. 5M-1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with pin #1 and continue counterclockwise to pin #20 when viewed from the top.

Package Outlines

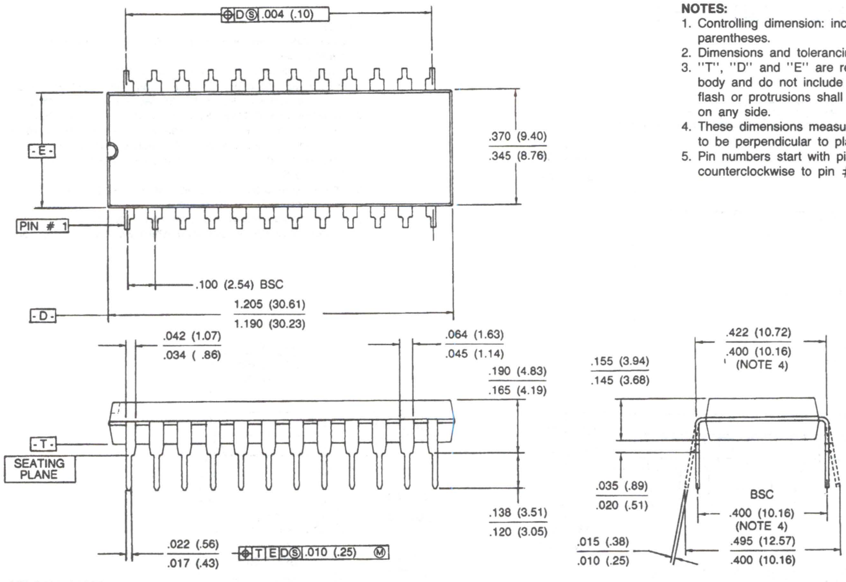
24-PIN PLASTIC DIP (300mil-wide)


D Ⓢ .004 (.10)
-E-
PIN #1
.255 (6.48)
.245 (6.22)
.100 2.54 (BSC)
-D-
1.256 (31.90)
1.240 (31.50)
.064 (1.63)
.045 (1.14)
-T-
SEATING PLANE
.160 (4.06)
.135 (3.43)
.138 (3.51)
.120 (3.05)
.022 (.56)
.017 (.43)
T E D Ⓢ .010 (.25) Ⓜ
.322 (8.18)
.300 (7.62)
(NOTE 5)
.125 (3.18)
.115 (2.92)
.035 (.89)
.020 (.51)
BSC
.300 (7.62)
(NOTE 5)
.015 (.38)
.010 (.25)
.395 (10.03)
.300 (7.62)
853-0410 81236
PO00320S

NOTES:

1. Controlling dimension: inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC specification MS-001-AR for standard dual in-line (DIP) package .300 inch row spacing (PLASTIC) 24 leads (issue B. 7/85)
3. Dimensions and tolerancing per ANSI Y14. 5M-1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with pin #1 and continue counterclockwise to pin #24 when viewed from the top.

24-PIN PLASTIC DIP (400mil-wide)

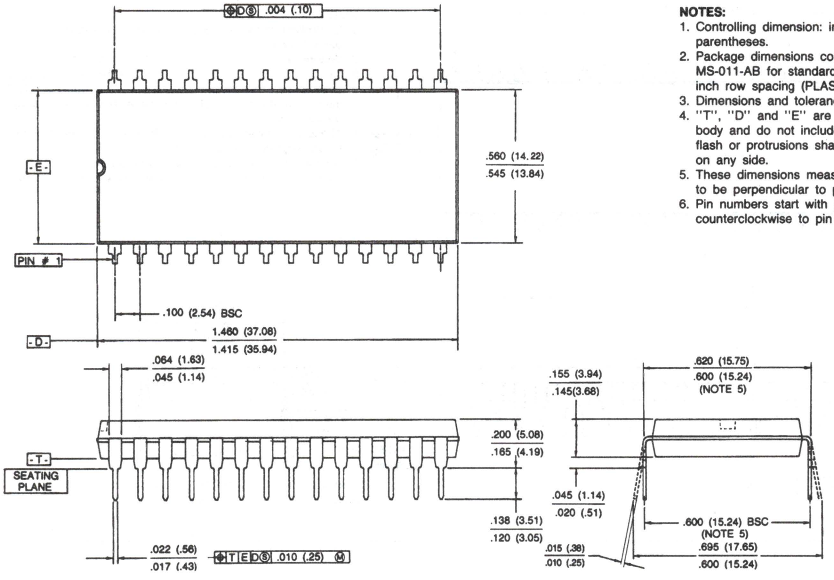

D Ⓢ .004 (.10)
-E-
PIN # 1
.370 (9.40)
.345 (8.76)
.100 (2.54) BSC
-D-
1.205 (30.61)
1.190 (30.23)
.042 (1.07)
.034 (.86)
.064 (1.63)
.045 (1.14)
.190 (4.83)
.165 (4.19)
-T-
SEATING PLANE
.138 (3.51)
.120 (3.05)
.022 (.56)
.017 (.43)
T E D Ⓢ .010 (.25) Ⓜ
.422 (10.72)
.400 (10.16)
(NOTE 4)
.155 (3.94)
.145 (3.68)
.035 (.89)
.020 (.51)
BSC
.400 (10.16)
(NOTE 4)
.015 (.38)
.010 (.25)
.495 (12.57)
.400 (10.16)
853-0411 81237
PO00351S

NOTES:

1. Controlling dimension: inches. Metric are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14. 5M-1982.
3. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #24 when viewed from the top.

Package Outlines

28-PIN PLASTIC DIP

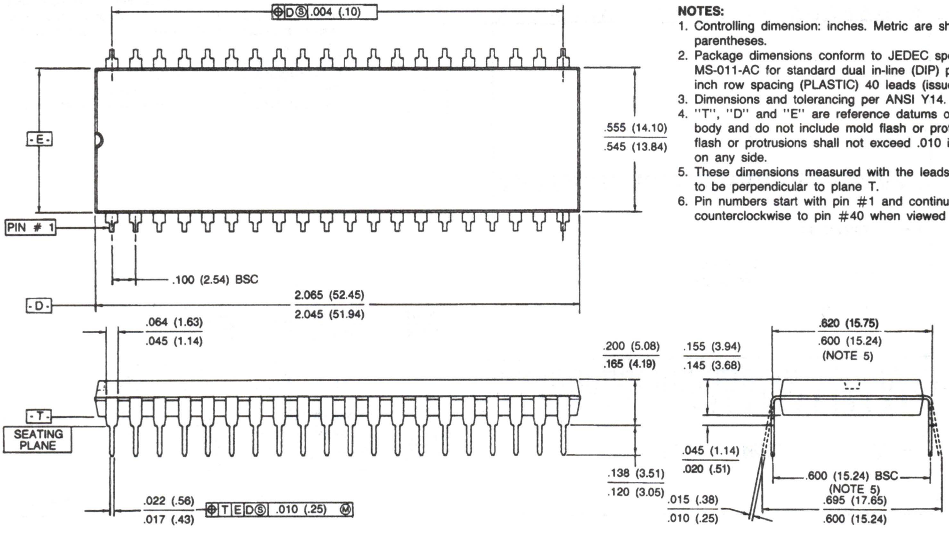

.004 (.10)
-E-
PIN # 1
.560 (14.22)
.545 (13.84)
.100 (2.54) BSC
-D-
1.460 (37.08)
1.415 (35.94)
.064 (1.63)
.045 (1.14)
.155 (3.94)
.145(3.68)
.620 (15.75)
.600 (15.24)
(NOTE 5)
.200 (5.08)
.165 (4.19)
-T-
SEATING PLANE
.045 (1.14)
.020 (.51)
.138 (3.51)
.120 (3.05)
.600 (15.24) BSC
(NOTE 5)
.695 (17.65)
.600 (15.24)
.022 (.56)
.017 (.43)
T E D .010 (.25)
.015 (.38)
.010 (.25)

NOTES:
1. Controlling dimension: inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC specification MS-011-AB for standard dual in-line (DIP) package .600 inch row spacing (PLASTIC) 28 leads (issue B. 7/85)
3. Dimensions and tolerancing per ANSI Y14. 5M-1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with pin #1 and continue counterclockwise to pin #28 when viewed from the top.

853-0413 84099

PO00373S

40-PIN PLASTIC DIP


.004 (.10)
-E-
PIN # 1
.555 (14.10)
.545 (13.84)
.100 (2.54) BSC
-D-
2.065 (52.45)
2.045 (51.94)
.064 (1.63)
.045 (1.14)
.200 (5.08)
.165 (4.19)
.155 (3.94)
.145 (3.68)
.620 (15.75)
.600 (15.24)
(NOTE 5)
-T-
SEATING PLANE
.045 (1.14)
.020 (.51)
.138 (3.51)
.120 (3.05)
.600 (15.24) BSC
(NOTE 5)
.695 (17.65)
.600 (15.24)
.022 (.56)
.017 (.43)
T E D .010 (.25)
.015 (.38)
.010 (.25)

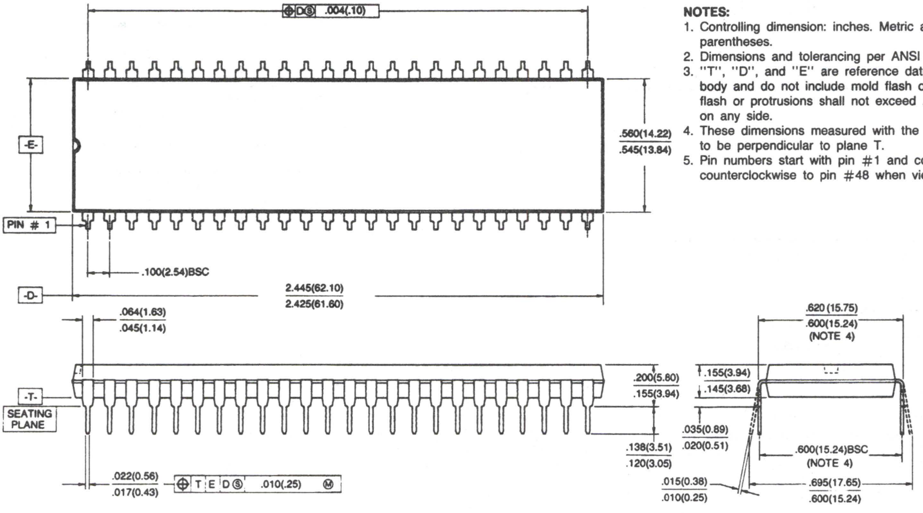
NOTES:
1. Controlling dimension: inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC specification MS-011-AC for standard dual in-line (DIP) package .600 inch row spacing (PLASTIC) 40 leads (issue B. 7/85)
3. Dimensions and tolerancing per ANSI Y14. 5M-1982.
4. "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with pin #1 and continue counterclockwise to pin #40 when viewed from the top.

853-0415 84971

PO00382S

Package Outlines

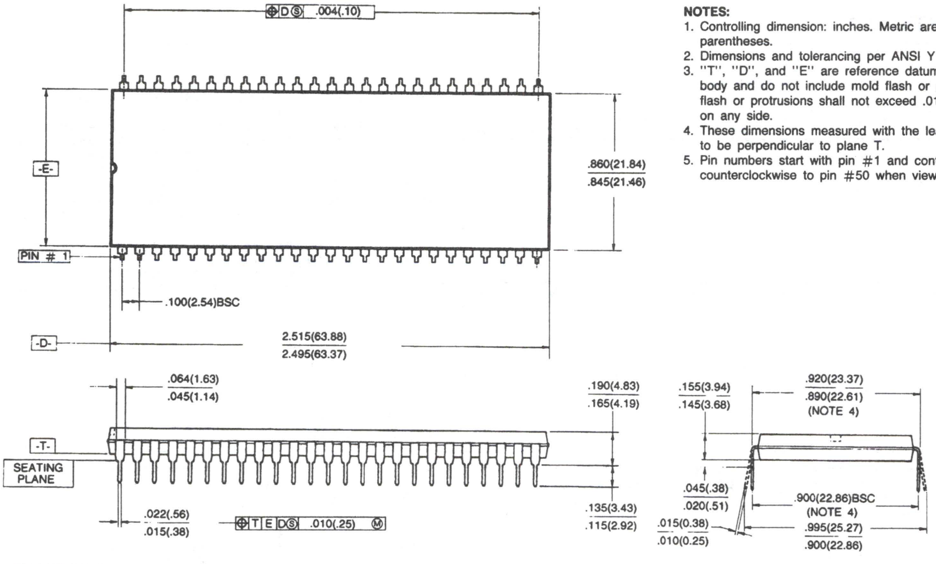
48-PIN PLASTIC DIP


⊕ D Ⓢ .004(.10)
-E-
PIN # 1
.560(14.22)
.545(13.84)
.100(2.54)BSC
-D-
2.445(62.10)
2.425(61.60)
.064(1.63)
.045(1.14)
-T-
SEATING PLANE
.200(5.80)
.155(3.94)
.138(3.51)
.120(3.05)
.022(0.56)
.017(0.43)
⊕ T E D Ⓢ .010(.25) Ⓜ
.620(15.75)
.600(15.24)
(NOTE 4)
.155(3.94)
.145(3.68)
.035(0.89)
.020(0.51)
.600(15.24)BSC
(NOTE 4)
.015(0.38)
.010(0.25)
.695(17.65)
.600(15.24)
853-0416 84099
PO00141S

NOTES:

1. Controlling dimension: inches. Metric are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #48 when viewed from the top.

50-PIN PLASTIC DIP

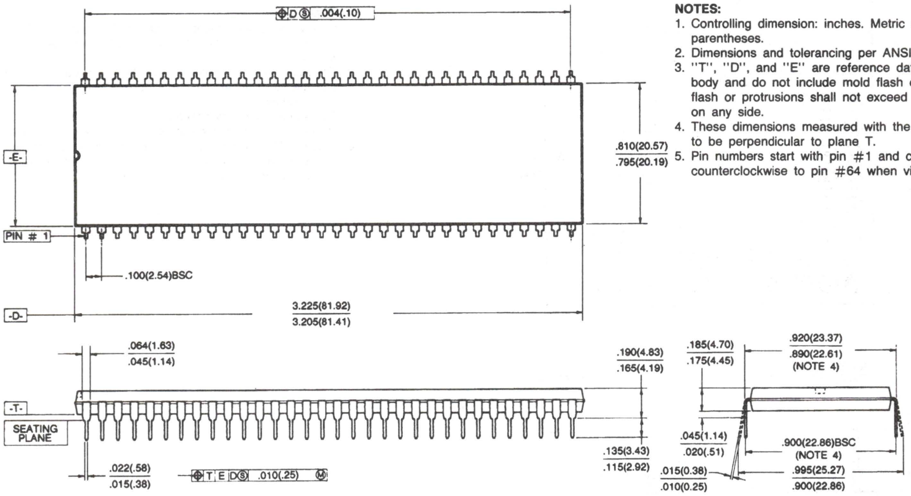

⊕ D Ⓢ .004(.10)
-E-
PIN # 1
.860(21.84)
.845(21.46)
.100(2.54)BSC
-D-
2.515(63.88)
2.495(63.37)
.064(1.63)
.045(1.14)
-T-
SEATING PLANE
.190(4.83)
.165(4.19)
.135(3.43)
.115(2.92)
.022(.56)
.015(.38)
⊕ T E D Ⓢ .010(.25) Ⓜ
.155(3.94)
.145(3.68)
.045(.38)
.020(.51)
.015(0.38)
.010(0.25)
.920(23.37)
.890(22.61)
(NOTE 4)
.900(22.86)BSC
(NOTE 4)
.995(25.27)
.900(22.86)
853-0417 81243
PO00151S

NOTES:

1. Controlling dimension: inches. Metric are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #50 when viewed from the top.

Package Outlines

64-PIN PLASTIC DIP


.004(.10)
D
-E-
PIN # 1
.100(2.54)BSC
-D-
3.225(81.92)
3.205(81.41)
.810(20.57)
.795(20.19)
.064(1.63)
.045(1.14)
-T-
SEATING PLANE
.022(.58)
.015(.38)
T E D .010(.25)
.190(4.83)
.165(4.19)
.135(3.43)
.115(2.92)
.185(4.70)
.175(4.45)
.045(1.14)
.020(.51)
.015(0.38)
.010(0.25)
.920(23.37)
.890(22.61)
(NOTE 4)
.900(22.86)BSC
(NOTE 4)
.995(25.27)
.900(22.86)

NOTES:

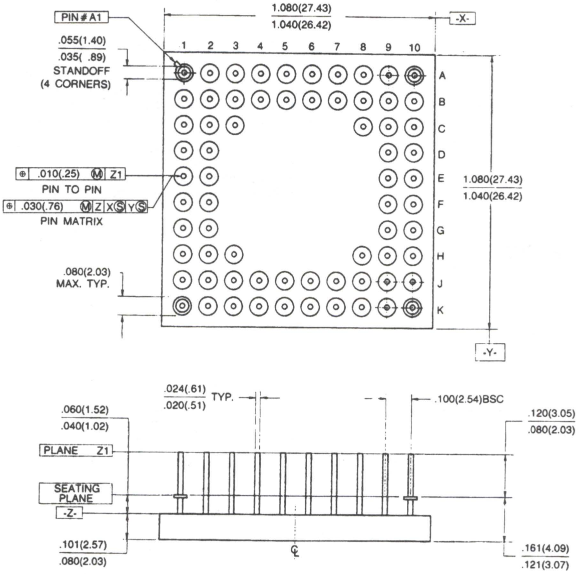
1. Controlling dimension: inches. Metric are shown in parentheses.
2. Dimensions and tolerancing per ANSI Y14.5M – 1982.
3. "T", "D", and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010 inch (.25mm) on any side.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with pin #1 and continue counterclockwise to pin #64 when viewed from the top.

853-0414 81240

PO00160S

Package Outlines

68-PIN GRID ARRAY



PIN # A1
1.080(27.43)
1.040(26.42)
-X-
.055(1.40)
.035(.89)
STANDOFF
(4 CORNERS)
1 2 3 4 5 6 7 8 9 10
A B C D E F G H J K
⊕ .010(.25) Ⓜ Z1
PIN TO PIN
⊕ .030(.76) Ⓜ Z X Ⓢ Y Ⓢ
PIN MATRIX
1.080(27.43)
1.040(26.42)
.080(2.03)
MAX. TYP.
-Y-
.024(.61)
.020(.51)
TYP.
.100(2.54)BSC
.060(1.52)
.040(1.02)
.120(3.05)
.080(2.03)
PLANE Z1
SEATING PLANE
-Z-
.101(2.57)
.080(2.03)
℄
.161(4.09)
.121(3.07)

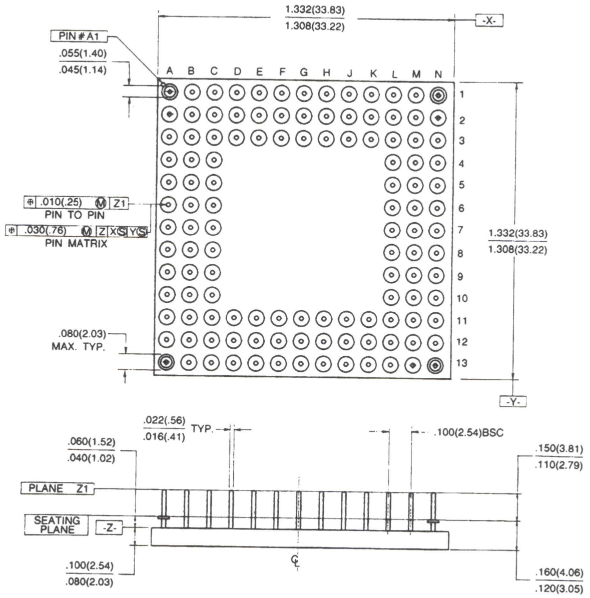
NOTES:

1. Dimensioning and tolerancing per ANSI Y14.5M – 1982
2. Terminal position designation per JEDEC publication 95. Section 2.2 – 5.
3. The total body thickness includes both the package body and the lid in either the cavity up or cavity down configuration.
4. The seating plane is defined by the standoffs.
5. Datum "Z" reference plane of pin to package interface.
6. Datum "ZI" reference plane is the plane of the pin tips.
7. Pin A1 corner identification will be visible on both the top and bottom surfaces of the package.

853-0170 83946

PO00740S

120-PIN GRID ARRAY



1.332(33.83)
1.308(33.22)
-X-
PIN # A1
.055(1.40)
.045(1.14)
A B C D E F G H J K L M N
1 2 3 4 5 6 7 8 9 10 11 12 13
⊕ .010(.25) Ⓜ Z1
PIN TO PIN
⊕ .030(.76) Ⓜ Z X Ⓢ Y Ⓢ
PIN MATRIX
1.332(33.83)
1.308(33.22)
.080(2.03)
MAX. TYP.
-Y-
.022(.56)
.016(.41)
TYP.
.100(2.54)BSC
.060(1.52)
.040(1.02)
.150(3.81)
.110(2.79)
PLANE Z1
SEATING PLANE
-Z-
.100(2.54)
.080(2.03)
℄
.160(4.06)
.120(3.05)

NOTES:

1. Dimensioning and tolerancing per ANSI Y14.5M – 1982
2. Terminal position designation per JEDEC publication 95. Section 2.2 – 5.
3. The total body thickness includes both the package body and the lid in either the cavity up or cavity down configuration.
4. The seating plane is defined by the standoffs.
5. Datum "Z" reference plane is the plane of pin to package interface.
6. Datum "ZI" reference plane is the plane of the pin tips.
7. Pin A1 corner identification will be visible on both the top and bottom surfaces of the package.

853-0853 83946

PO00790S

Signetics

Section 5
Sales Offices

Microprocessor Products

Sales Offices

SIGNETICS HEADQUARTERS
811 East Arques Avenue
P.O. Box 3409
Sunnyvale,
California 94088-3409
Phone: (408) 991-2000

ALABAMA
Huntsville
Phone: (205) 830-4001

ARIZONA
Phoenix
Phone: (602) 265-4444

CALIFORNIA
Canoga Park
Phone: (818) 340-1431

Irvine
Phone: (714) 833-8980
(213) 588-3281

Los Angeles
Phone: (213) 670-1101

San Diego
Phone: (619) 560-0242

Sunnyvale
Phone: (408) 991-3737

COLORADO
Aurora
Phone: (303) 751-5011

FLORIDA
Clearwater
Phone: (813) 796-7086

Ft. Lauderdale
Phone: (305) 486-6300

GEORGIA
Atlanta
Phone: (404) 953-0067

ILLINOIS
Itasca
Phone: (312) 250-0050

INDIANA
Kokomo
Phone: (317) 453-6462

KANSAS
Overland Park
Phone: (913) 469-4005

MASSACHUSETTS
Littleton
Phone: (617) 486-8411

MICHIGAN
Farmington Hills
Phone: (313) 476-1610

MINNESOTA
Edina
Phone: (612) 835-7455

NEW JERSEY
Parsippany
Phone: (201) 334-4405

NEW YORK
Hauppauge
Phone: (516) 348-7877

Wappingers Falls
Phone: (914) 297-4074

NORTH CAROLINA
Cary
Phone: (919) 481-0400

OHIO
Worthington
Phone: (614) 888-7143

OREGON
Portland
Phone: (503) 297-5592

PENNSYLVANIA
Plymouth Meeting
Phone: (215) 825-4404

TENNESSEE
Greeneville
Phone: (615) 639-0251

TEXAS
Austin
Phone: (512) 339-9944

Richardson
Phone: (214) 644-3500

CANADA
SIGNETICS CANADA, LTD.
Etobicoke, Ontario
Phone: (416) 626-6676

Nepean, Ontario
Signetics, Canada, Ltd.
Phone: (613) 726-9576

REPRESENTATIVES

ARIZONA
Scottsdale
Thom Luke Sales, Inc.
Phone: (602) 941-1901

CALIFORNIA
Santa Clara
Magna Sales
Phone: (408) 727-8753

CONNECTICUT
Brookfield
M & M Associates
Phone: (203) 775-6888

FLORIDA
Clearwater
Sigma Technical Sales
Phone: (813) 791-0271

Ft. Lauderdale
Sigma Technical Sales
Phone: (305) 731-5995

ILLINOIS
Hoffman Estates
Micro-Tex, Inc.
Phone: (312) 382-3001

INDIANA
Indianapolis
Mohrfield Marketing Inc.
Phone: (317) 546-6969

KANSAS
Kansas City
B.C. Electronic Sales
Phone: (913) 342-1211

MARYLAND
Glen Burnie
Third Wave Solutions, Inc.
Phone: (301) 787-0220

MASSACHUSETTS
Needham Heights
Kanan Associates
Phone: (617) 449-7400
Com-Sales, Inc.
Phone: (617) 444-8071

MICHIGAN
Bloomfield Hills
Enco Marketing
Phone: (313) 642-0203

MINNESOTA
Eden Prairie
High Technology Sales
Phone: (612) 944-7274

MISSOURI
St. Louis
B.C. Electronic Sales
Phone: (314) 521-6683

NEW JERSEY
East Hanover
Emtec Sales, Inc.
Phone: (201) 428-0600

NEW MEXICO
Albuquerque
F.P. Sales
Phone: (505) 345-5553

NEW YORK
Ithaca
Bob Dean, Inc.
Phone: (607) 257-1111

OHIO
Cleveland
Covert & Newman
Phone: (216) 663-3331
Dayton
Covert & Newman
Phone: (513) 439-5788
Worthington
Covert & Newman
Phone: (614) 888-2442

OKLAHOMA
Tulsa
Jerry Robinson and
Associates
Phone: (918) 665-3562

OREGON
Hillsboro
Western Technical Sales
Phone: (503) 640-4621

PENNSYLVANIA
Pittsburgh
Covert & Newman
Phone: (412) 531-2002

Willow Grove
Delta Technical Sales Inc.
Phone: (215) 657-7250

TEXAS
Houston
OM Sales
Phone: (713) 789-4426

UTAH
Salt Lake City
Electrodyne
Phone: (801) 486-3801

WASHINGTON
Bellevue
Western Technical Sales
Phone: (206) 641-3900

Spokane
Western Technical Sales
Phone: (509) 922-7600

WISCONSIN
Waukesha
Micro-Tex, Inc.
Phone: (414) 542-5352

CANADA
Burnaby, British Columbia
Tech-Trek, Ltd.
Phone: (604) 439-1367

Mississauga, Ontario
Tech-Trek, Ltd.
Phone: (416) 238-0366

Nepean, Ontario
Tech-Trek, Ltd.
Phone: (613) 726-9562

Richmond, British Columbia
Tech-Trek, Ltd.
Phone: (604) 271-3149

Ville St. Laurent, Quebec
Tech-Trek, Ltd.
Phone: (514) 337-7540

Sales Offices

DISTRIBUTORS

Contact one of our local distributors:
Anthem Electronics
Arrow Electronics
Avnet Electronics
Hamilton/Avnet Electronics
Lionex Corporation
Schweber Electronics
Summit Distributors
Quality Components
Wyle LEMG
Zentronics, Ltd.

FOR SIGNETICS PRODUCTS WORLDWIDE:

ARGENTINA
Philips Argentina S.A.
Buenos Aires
Phone: 54-1-541-7141

AUSTRALIA
Philips Electronic Components and Materials, Ltd.
Artarmon, N.S.W.
Phone: 61-2-439-3322

AUSTRIA
Osterrichische Philips Bauelemente
Wien
Phone: 43-222-62-91-11

BELGIUM
N.V. Philips & MBLE
Bruxelles
Phone: 32-2-5-23-00-00

BRAZIL
Philips Do Brasil, Ltda.
Sao Paulo
Phone: 55-11-211-2600

CHILE
Philips Chilena S.A.
Santiago
Phone: 56-02-077-3816

COLOMBIA
Iprelenso, Ltda.
Bogota
Phone: 57-1-2497624

Effective 2-24-87

DENMARK
Miniwatt A/S
Copenhagen S
Phone: 45-1-54-11-33

FINLAND
Oy Philips Ab
Helsinki
Phone: 358-0-172-71

FRANCE
R.T.C. La Radiotechnique-Compelec
Paris
Phone: 33-1-43-38-80-00

GERMANY
Valvo
Hamburg
Phone: 49-40-3-296-1

GREECE
Philips Hellenique S.A.
Athens
Phone: 30-1-9-21-5111

HONG KONG
Philips Hong Kong, Ltd.
Kwai Chung
Phone: 852-0-245-121

INDIA
Peico Electronics & Elect. Ltd.
Bombay
Phone: 91-22-493-8721

INDONESIA
P.T. Philips-Ralin Electronics
Jakarta Selatan
Phone: 62-21-512-572

IRELAND
Philips Electrical Ltd.
Dublin
Phone: 353-1-69-33-55

ISRAEL
Rapac Electronics, Ltd.
Tel Aviv
Phone: 972-3-477115

ITALY
Philips S.p.A.
Milano
Phone: 39-2-67-52-1

JAPAN
Nikon Philips Corp.
Tokyo
Phone: 81-3-448-5617
Signetics Japan Ltd.
Phone: 81-3-230-1521/2

KOREA
Philips Industries, Ltd.
Seoul
Phone: 82-2-794-5011/2/3/4/5

MEXICO
Electronica S.A. de C.V.
Toluca
Phone: (721) 613-00

NETHERLANDS
Philips Nederland B.V.
Eindhoven
Phone: 31-40-793-333

NEW ZEALAND
Philips New Zealand Ltd.
Auckland
Phone: 64-9-605914

NORWAY
Norsk A/S Philips
Oslo
Phone: 47-2-68-02-00

PERU
Cadesa
Lima
Phone: 51-14-319253

PHILIPPINES
Philips Industrial Dev., Inc.
Makati
Phone: 63-2-868951-9

PORTUGAL
Philips Portuguesa SARL
Lisbon
Phone: 351-1-65-71-85

SINGAPORE
Philips Project Dev. Pte., Ltd.
Singapore
Phone: 65-350-2000

SOUTH AFRICA
E.D.A.C. (PTY), Ltd.
Joubert Park
Phone: 27-11-402-4600

SPAIN
Miniwatt S.A.
Barcelona
Phone: 34-3-301-63-12

SWEDEN
Philips Komponenter A.B.
Stockholm
Phone: 46-8-782-10-00

SWITZERLAND
Philips A.G.
Zurich
Phone: 41-1-488-2211

TAIWAN
Philips Taiwan, Ltd.
Taipei
Phone: 886-2-712-0500

THAILAND
Philips Electrical Co. of Thailand Ltd.
Bangkok
Phone: 66-2-233-6330-9

TURKEY
Turk Philips Ticaret A.S.
Istanbul
Phone: 90-11-43-59-10

UNITED KINGDOM
Mullard, Ltd.
London
Phone: 44-1-580-6633

UNITED STATES
Signetics International Corp.
Sunnyvale, California
Phone: (408) 991-2000

URUGUAY
Luzilectron, S.A.
Montevideo
Phone: 598-91-56-41/42/43/44

VENEZUELA
Magnetica, S.A.
Caracas
Phone: 58-2-241-7509

Section 6
Alphanumeric Index

Microprocessor Products

INDEX

Alphanumeric Index 6-3

Signetics

Alphanumeric Index

Microprocessor Products

AN4011	An 8X401-Based Raster Graphics System	3-279
AN408	SC83C451 Operation of Port 6	3-123
FORTRESS 8X401	Cross Assembler Program	3-328
I^2C Bus Specification		3-111
I/O Port Programmer		3-332
MCCAP 8X300/8X305	Cross Assembler Program	3-327
PCB5010/11	Single-Chip Digital Signal Processors	3-106
SCB2673	Video Attributes Controller (VAC)	2-132
SCB2675	Color/Monochrome Attributes Controller (CMAC)	2-177
SCB2675T	Turbo Color/Monochrome Attributes Controller (Turbo-CMAC)	2-188
SCB2677	Video Attributes Controller (VAC)	2-199
SCB68154	Interrupt Generator	2-471
SCB68155	Interrupt Handler	2-482
SCB68171	Very Little Serial Interface Chip (VLSIC)	2-499
SCB68172	VMEbus Controller (BUSCON)	2-504
SCB68430	Direct Memory Access Interface (DMAI)	2-544
SCC2691	Universal Asynchronous Receiver/Transmitter (UART)	2-230
SCC2692	Dual Asynchronous Receiver/Transmitter (DUART)	2-247
SCC2698	Octal Universal Asynchronous Receiver/Transmitter (Octal UART)	2-266
SCC63484	Advanced CRT Controller (ACRTC)	2-285
SCC68173	VMSbus Controller (VMSCON)	2-530
SCC68692	Dual Asynchronous Receiver/Transmitter (DUART)	2-632
SCN2641	Asynchronous Commmunications Interface	2-3
SCN2651	Programmable Communications Interface (PCI)	2-16
SCN2652/68652	Multi-Protocol Communications Controller (MPCC)	2-32
SCN2653/68653	Polynomial Generator Checker (PGC)	2-52
SCN2661/68661	Enhanced Programmable Communications Interface (EPCI)	2-70
SCN2671	Programmable Keyboard and Communication Controller (PKCC)	2-87
SCN2672	Programmable Video Timing Controller (PVTC)	2-109
SCN2674	Advanced Video Display Controller (AVDC)	2-145
SCN2681	Dual Asynchronous Receiver/Transmitter (DUART)	2-211
SCN68000	16-/32-Bit Microprocessor	2-340
SCN68010	16-Bit Virtual Memory Microprocessor	2-402
SCN68562	Dual Universal Serial Communications Controller (DUSCC)	2-565
SCN68681	Dual Asynchronous Receiver/Transmitter (DUART)	2-612
SCN8031AH/SCN8051AH	Single-Chip 8-Bit Microcontroller	3-18
SCN8032AH/SCN8052AH	Single-Chip 8-Bit Microcontroller	3-29
SCN8049 Series	Single-Chip 8-Bit Microcontroller	3-3
SC80C31B/SC80C51B	CMOS Single-Chip 8-Bit Microcontroller	3-40
SC80C451/SC83C451	I/O Expanded CMOS Microcontroller	3-54
SC80C552/SC83C552	CMOS Automotive and Industrial Microcontroller	3-73
SC80C652/SC83C652	CMOS Single-Chip 8-Bit Microcontroller	3-26
SC83C751	CMOS Single-Chip 8-Bit Microcontroller	3-71
SC87C451	CMOS Single-Chip 8-Bit EPROM Microcontroller	3-70
SC96AH Series	Single-Chip 16-Bit Microcontrollers	3-79
SC96BH Series	Single-Chip 16-Bit Microcontrollers	3-103
8X300 KT1SK	8X305 Prototyping System	3-329
8X305 Development System	American Automation EZ-PRO	3-331
8X305	Microcontroller	3-134
8X310	Interrupt Control Coprocessor	3-159
8X320	Bus Interface Register Array	3-172
8X330	Floppy Disk Formatter/Controller	3-180
8X353	Bipolar RAM (32 × 8)	3-197
8X355	LIFO Stack Memory (32 × 8)	3-205
8X360	Memory Address Director (MAD)	3-214
8X371	8-Bit Latched Bidirectional I/O Port	3-223
8X372/8X376	Addressable/Bidirectional I/O Port	3-230
8X374	Addressable/Bidirectional I/O Port with Parity	3-239
8X382	4-Input/4-Output Addressable I/O Port	3-249
8X401	Microcontroller	3-259

Alphanumeric Index

8X411	Priority Interrupt Encoder	3-304
8X450	RAM	3-306
8X470	I/O Port	3-313

NOTES

NOTES

Signetics
a subsidiary of U.S. Philips Corporation

Signetics Corporation
811 E. Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088-3409
Telephone 408/991-2000

98-8000-000

© Copyright 1987 Signetics Corporation
Printed in USA

AIS/MP60MCR2187
1280 pages